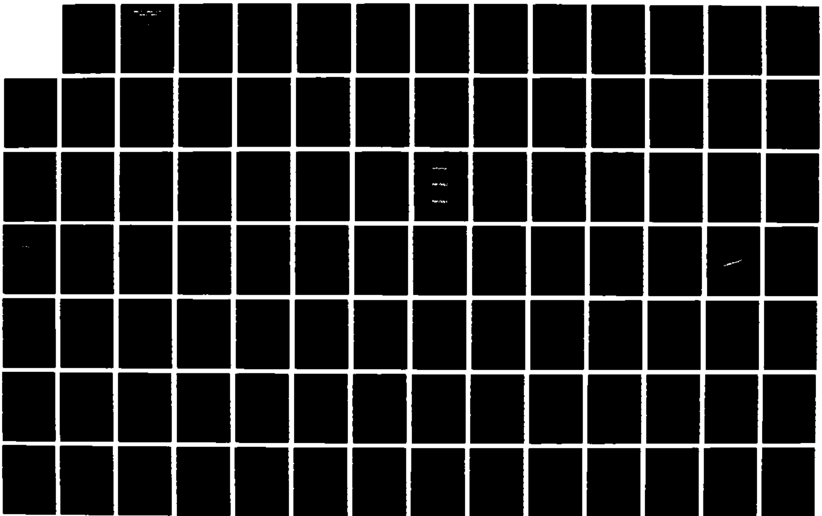
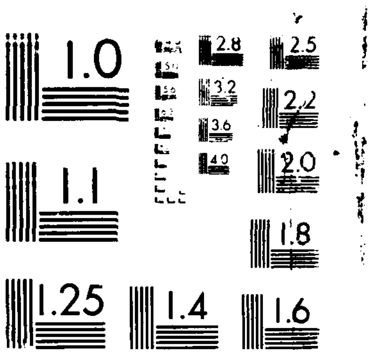
AD-A195 193 PC SOFTWARE FOR THE TEACHING OF DIGITAL SIGNAL PROCESSING(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA Y KATZIR MAR 88 1/2

UNCLASSIFIED F/G 12/5 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

PC SOFTWARE FOR THE TEACHING OF
DIGITAL SIGNAL PROCESSING

by

Yoel Katzir

March 1988

Thesis Advisor C. W. Therrien

Approved for public release; distribution is unlimited.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

A/ 25 28

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
PC SOFTWARE FOR THE TEACHING OF DIGITAL SIGNAL PROCESSING

12. PERSONAL AUTHOR(S)
Katzir, Yoel

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Master's Thesis	FROM ____ TO ____	March 1988	180

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	digital signal processing, APL Language, function listings, workspace

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The Electrical and Computer Engineering Department at the Naval Postgraduate School has a need for additional software to be used in instructing students studying signal processing. This software will be used in a PC lab or at home.

This thesis provides a set of disks written in APL (A Programming Language) which allows the user to input arbitrary signals from a disk, to perform various signal processing operations, to plot the results, and to save them without the need for complicated programming.

The software is in the form of a digital signal processing "toolkit." The user can select functions which can operate on the signals and interactively apply them in any order. The user can also easily develop new functions and include them in the "toolkit."

The thesis includes brief discussions about the library workspaces, a user manual, function listings with examples of their use, and an application paper. The software is modular and can be expanded by adding additional sets of functions.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL C. W. Therrien, Code 62Ti	22b. TELEPHONE (Include Area Code) 408-646-3347 / 22c. OFFICE SYMBOL 62Ti

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

☆ U.S. Government Printing Office: 1986—606-24.

Approved for public release; distribution is unlimited

PC Software for the Teaching of Digital Signal Processing

by

Yoel Katzir
Major, Israeli Air Force
B.S.EE, Ben-Gurion University of the Negav, Israel, 1977

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1988

Author: Y. Katzir
Yoel Katzir

Approved by: C. W. Therrien
C. W. Therrien, Thesis Advisor

R. D. Strum
R. D. Strum, Second Reader

J. P. Powers, Chairman, Department of
Electrical and Computer Engineering

G. Schacher
Gordon E. Schacher,
Dean of Science and Engineering

ABSTRACT

The Electrical and Computer Engineering Department at the Naval Postgraduate School has a need for additional software to be used in instructing students studying signal processing. This software will be used in a PC lab or at home.

This thesis provides a set of disks written in APL (A Programming Language) which allows the user to input arbitrary signals from a disk, to perform various signal processing operations, to plot the results, and to save them without the need for complicated programming.

The software is in the form of a digital signal processing "toolkit." The user can select functions which can operate on the signals and interactively apply them in any order. The user can also easily develop new functions and include them in the "toolkit."

The thesis includes brief discussions about the library workspaces, a user manual, function listings with examples of their use, and an application paper. The software is modular and can be expanded by adding additional sets of functions.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

The Electrical and Computer Engineering Department at the Naval Postgraduate School has a need for signal processing software packages for use by students studying digital signal processing (DSP). The purpose of this thesis is to provide a set of tools in the form of APL workspaces on floppy disks that students can use at home or at school on IBM-PC-compatible computers for solving digital signal processing problems. Workspace is a collection of functions and data that are placed together in order to do any job.

The software supports the following currently offered courses in DSP:

1. EC 2400-Discrete Systems
2. EC 3400-Introduction to Digital Signal Processing
3. EC 3410-Introduction to Discrete Time Random Processes
4. EC 4430-Advanced Signal Processing and Spectral Estimation
5. EC 4440-Multidimensional Digital Signal Processing

Specific APL workspaces were developed for EC3400, EC3410, and EC4440. The software uses STSC Corporation's APL*PLUS PC and STATGRAPHICS systems which are both site licensed to the Naval Postgraduate School. APL*PLUS PC is STSC's version of the APL language for the IBM-PC and compatibles. Similar versions of the APL language are also available from STSC for other computers. STATGRAPHICS (Statistical Graphics System) is a PC software package integrating a variety of statistical functions with color graphics. STATGRAPHICS is written in APL (specifically APL*PLUS PC) and offers direct access to the

host language. Currently STATGRAPHICS is available only for the IBM-PC and compatibles.

The DSP software developed in this thesis is in the form of a digital signal processing "toolkit." The user can select functions to operate on the signals and interactively apply them in any order. The user can also easily develop new functions and include them in the "toolkit." The tools are provided on a set of floppy disks that users can carry to their own PC machine. The sets of disks contain:

1. The APL*PLUS Executive Program
2. STATGRAPHICS disks
3. UTILITY workspace
4. EC3400 workspace
5. EC3410 workspace
6. EC4440 workspace

The main products of the research are the disks that include the signal processing library workspace, and the documentation provided in this thesis. A typical library contains functions such as: Fast Fourier Transform (FFT), Inverse FFT, Sine, Cosine, Sample Autocorrelation, Mean, and other similar functions. In developing the DSP software the following requirements were met:

1. Ability to input/output from/to a disk data file in free format. This allows convenient communication with DOS and with programs written in other languages or on other computer systems.
2. Ability to provide for graphic output to represent signals, frequency responses, and other functions including 2-D, 3-D, and contour plotting.
3. Ability to perform filtering, convolution, and other common signal processing operations in the signal (time/space) domain.

4. Ability to compute transfer functions, to compute frequency responses, and generally to perform complex arithmetic and other common frequency domain operations.

5. Ability to generate random signals.

6. Ability to compute statistical characteristics of the signals including correlation functions, spectra, and prediction errors.

The user can combine functions in any desired order to solve DSP computer assignments and can concentrate on the signal processing exercises without the need to do complicated programming.

Although the intent was to develop the signal processing toolkit specifically for the IBM-PC and compatibles, the system is portable. STSC has APL available on IBM mainframes, Digital Equipment Corporation VAX, (under UNIX and VMS), Apple MacIntosh, and others. These DSP workspaces are portable to all of these systems with few, if any, changes. In addition, the DSP workspaces are easily carried over to other APL systems with minor changes to only some of the most basic functions; bits and pieces of the software have existed for some time under VS-APL on the IBM mainframe and under UNIX 4.3 bsd.

The remainder of the thesis is organized as follows.

Chapter II contains a brief description of the APL language and the STATGRAPHICS software system. Chapter III deals with the DSP library and discusses the application of the four workspaces. The Utility workspace includes general functions such as GETDATA and PUTDATA for the data input/output to DOS; the EC3400 workspace includes basic one-dimensional DSP functions; the EC3410 workspace includes one-dimensional DSP functions for statistical signal processing; and the EC4440 workspace includes multidimensional DSP functions. Chapter IV outlines the conclusions, and the benefits of using the entire package as a software

toolkit. Appendix A provides a simple user manual for the software package. Appendix B provides a function library with a brief description of each function. Appendix C contains function listings and samples of their use. Appendix D furnishes samples of computer assignments and their solutions using the tools developed in this thesis.

II. APL AND STATGRAPHICS ENVIRONMENT

The signal processing software is written in APL and uses the APL*PLUS PC interpreter and the STATGRAPHICS software system. Although STATGRAPHICS provides many statistical functions in itself, only the plotting functions from STATGRAPHICS are used. Following is a brief description of APL and the STATGRAPHICS system.

A. APL [1]

Originally APL began as a notation to express mathematical procedures. Its originator, Dr. Kenneth Iverson, published his notation in 1962 as **A Programming Language** from which the name "APL" is derived [2].

One of the strongest advantages of APL is its ability to handle complicated array data-structures without extensive programming. The language allows the definition of numerical items or a series of items with one symbol. Examples of data that can be represented are:

- SCALAR - A single value
- VECTOR - A group of scalars
- MATRIX - A group of vectors
- ARRAY - A group of matrices

The language makes it as easy to perform an arithmetic operation using a matrix or vector argument as to perform the operation using simple numbers. This characteristic is illustrated in the following example.

```
      M←ι6
      M
1 2 3 4 5 6
```

(define vectors with 6 numbers)

```
      N←2×ι6
      N
2 4 6 8 10 12
```

```
      N×M
2 8 18 32 50 72
```

(vector multiplication)

```
      M←3 3 ρι9
      M
1 2 3
4 5 6
7 8 9
```

(define matrices with 3 rows and 3 columns)

```
      N←3 3 ρ3×ι9
      N
 3  6  9
12 15 18
21 24 27
```

```
      N×M
  3  12  27
 48  75 108
147 192 243
```

(matrix arithmetic operation)

Notice that when two items are of the same dimensions, the arithmetic operation is performed element by element. Thus $N \times M$ is not the usual linear algebra product of matrices. The latter is realized by a composite operation denoted by $+.\times$.

```
      N+.×M
 90 108 126
198 243 288
306 378 450
```

Scalar arguments are automatically extended to be conformable with the arrays. Thus the statement $M \times 2$ yields

```
 2   4   6
 8  10  12
14  16  18
```

There is frequently no need for loops or similar mechanisms as in other languages such as Fortran. From the user's point of view the language performs the operation on the elements of the array in parallel without the need for loops.

Note also that there is no need for dimensions because APL carries the dimensions of an item along with its definitions.

APL provides a function to help the user to define or to determine the number of elements in the variable. The function represented by the Greek character "ρ", is called "shape," since it gives the form of the variable. On the previous page there is an example where we define the matrices M and N using the function "ρ". In order to find the shape of any variable, use the function "ρ" with the variable name following it. The following is an example.

```
      ρM
3  3
```

APL has a large variety of mathematical functions that can easily manipulate the data and can handle operations on a simple as well as on a complicated level. Thus, it is possible to place on a single line a combination of several functions which work together in a pipeline. All of this can be done interactively. The flexibility in manipulating array data easily makes the language well suited for single processing analysis. The following are a few examples:

```
      2 3 4 5 ∘.+ 6 ¯1 ¯3 4
 8  1 ¯1  6
 9  2  0  7
10  3  1  8
11  4  2  9
```

(outer product) used to define separable 2-D function $x(n1, n2) = x1(n1)x2(n2)$

```
      M←3 3 ρ1 3 5 7 9 0 8 6 4
      M
1 3 5
7 9 0
8 6 4
```

The domino operator (calculate M inverse)

```
      ⌹M
¯0.1818181818  ¯0.09090909091   0.2272727273
 0.1414141414   0.1818181818   ¯0.1767676768
 0.1515151515  ¯0.09090909091   0.06060606061

      C←11 3 6
```

(solve $Mx = C$ for x)

```
      C⌹M
¯0.9090909091 1.04040404 1.757575758
```

(check solution by multiplying constant vector by M inverse) $+.\times$ is matrix product

```
      (⌹M)+.×C
¯0.9090909091 1.04040404 1.757575758
```

APL uses symbols to represent its own built-in functions and, in addition, APL allows the user to define other functions. The solution to one problem consists of a collection of the programmer's functions, each one of which can be applied interactively by the user or can be organized to call another function. The advantage of this organization is the ease of debugging and that only a few lines of APL can do volumes of work.

APL has a concept called the workspace which is a collection of functions and data that are placed together in order to do a job. That collection is stored in the computer memory during an APL session and gives the operator immediate

feedback and accessibility, and the workspace can be saved on disk and loaded back into the memory at any time for future use.

B. STATGRAPHICS SOFTWARE PACKAGE [3]

STATGRAPHICS is a Statistical Graphics System written in APL for scientific and engineering applications and for data analysis. The software supports dot matrix printers, monochrome/color terminals, and pen plotters. It runs under the APL*PLUS PC interpreter version 5.0 or later, which allows the user to write his own functions. Using STATGRAPHICS in this mode requires familiarity with APL.

The system also has a menu that contains graphic functions, a set of data analysis and statistical procedures. This menu is illustrated in Figure 2.1. In addition, the menu contains data management and system utilities that allow the user to select the system environment, control color, plot dimensions, and so on. The most often used screens for DSP applications are:

1. Plotting functions (Figure 2.2)
2. Time Series Analysis (Figure 2.3)
3. Experimental Design (Figure 2.4)

STATGRAPHICS has options for modifying graphs and for on-line help.

To use STATGRAPHICS with APL the user needs to start APL*PLUS PC as usual and then load the STATGRAPHICS workspace from the start-up disk. By going back and forth from the APL environment to STATGRAPHICS, the user can process data and plot the results easily. This makes the combination of APL and STATGRAPHICS powerful and very convenient to use.

```
STATGRAPHICS Statistical Graphics System

DATA MANAGEMENT AND SYSTEM UTILITIES       TIME SERIES PROCEDURES
  A. Data Management                          L. Forecasting
  B. System Environment                       M. Quality Control
  C. Report Writer and Graphics Replay        N. Smoothing
  D. Plotter Interface                        O. Time Series Analysis

PLOTTING AND DESCRIPTIVE STATISTICS        ADVANCED PROCEDURES
  E. Plotting Functions                       P. Categorical Data Analysis
  F. Descriptive Methods                      Q. Multivariate Methods
  G. Estimation and Testing                   R. Nonparametric Methods
  H. Distribution Functions                   S. Sampling
  I. Exploratory Data Analysis                T. Experimental Design

ANOVA AND REGRESSION ANALYSIS              MATHEMATICAL AND USER PROCEDURES
  J. Analysis of Variance                     U. Mathematical Functions
  K. Regression Analysis                      V. Supplementary Operations

        Use cursor keys to highlight desired section.  Then press ENTER.
 1Help    2Edit    3Savscr 4Prtscr 5       6Go      7Vars    8Cmd     9Review 10Quit
INPUT     11/18/87  13:08 STATGRAPHICS Vers. 2.1
```

Figure 2.1. STATGRAPHICS Main Menu.

```
                     PLOTTING FUNCTIONS

             1.  X-Y Line and Scatterplots
             2.  Multiple X-Y Plots
             3.  X-Y-Z Line and Scatterplots
             4.  Multiple X-Y-Z Plots
             5.  Barcharts
             6.  Piecharts
             7.  Component Line Charts

        Use cursor keys to highlight desired procedure.  Then press ENTER.
 1Help    2Edit    3Savscr 4Prtscr 5       6Go      7Vars    8Cmd     9Review 10Quit
INPUT     11/18/87  13:10 STATGRAPHICS Vers. 2.1
```

Figure 2.2. The Plotting Function Menu.

```
TIME SERIES ANALYSIS

 1.  Horizontal Time Sequence Plot
 2.  Vertical Time Sequence Plot
 3.  Seasonal Subseries Plot
 4.  Autocorrelation Function
 5.  Partial Autocorrelation Function
 6.  Cross-Correlation Function
 7.  Simple or Seasonal Differencing
 8.  Mean or Trend Removal
 9.  Box-Cox Transformation
10.  Periodogram
11.  Integrated Periodogram
12.  Data Tapering
13.  Plotting vs. Fourier Frequencies
14.  Box-Jenkins ARIMA Modeling
15.  Cross-Correlation Matrix Plot

Use cursor keys to highlight desired procedure.  Then press ENTER.
1Help   2Edit   3Savscr 4Prtscr 5       6Go     7Vars    8Cmd    9Review 10Quit
INPUT   11/18/87  13:13 STATGRAPHICS Vers. 2.1
```

Figure 2.3. The Time Series Analysis Menu.

```
EXPERIMENTAL DESIGN

1.  Full and Fractional Factorials
2.  Central Composite Designs
3.  Alias Structure
4.  Response Surface Plotting

Use cursor keys to highlight desired procedure.  Then press ENTER.
1Help   2Edit   3Savscr 4Prtscr 5       6Go     7Vars    8Cmd    9Review 10Quit
```

Figure 2.4. The Experimental Design Menu.

III. THE DSP WORKSPACES AND APPLICATIONS

The following is the principle part of the thesis research. This chapter describes the workspaces that were developed by the author, and the way they can be applied.

A. UTILITY WORKSPACE

This workspace contains functions of general use that can be applied in combination with functions from other workspaces in this package. Appendix B contains the list of workspace functions and a brief discussion of each one.

Two of the most useful functions in the utilities are described and demonstrated below. The functions GETDATA and PUTDATA allow the user to transfer data from the APL environment to the DOS environment and vice versa. These functions use file system commands that are specific to STSC's implementation on the IBM- PC. As such, the exact APL code for GETDATA and PUTDATA is not always portable to APL on other systems. However, versions of GETDATA and PUTDATA can be written for any system. Versions are currently available for the IBM mainframe under VS/APL and for a version of APL under UNIX 4.3 bsd on the VAX. To the user, GETDATA and PUTDATA provide a common interface for communication with other programs.

The following is an example showing the transfer of data using the GETDATA and PUTDATA functions. Figure 3.1 describes two sets of data in an APL workspace. In order to transfer those sets to a file in the DOS environment the user needs to type the following commands:

```
X PUTDATA 'X.DAT'

Y PUTDATA 'Y.DAT'
```

X is 5×5 and Y is a $2 \times 4 \times 4$ array

```
                    X
 0.925  1.85    2.775  3.7     4.625
 5.55   6.475   7.4    8.325   9.25
10.175 11.1    12.025 12.95   13.875
14.8   15.725  16.65  17.575  18.5
19.425 20.35   21.275 22.2    23.125


         Y
  1E60     2E60     3E60     4E60
  5E60     6E60     7E60     8E60
  9E60     1E61    ¯1E44    ¯2E44
 ¯3E44    ¯4E44    ¯5E44    ¯6E44

 ¯7E44    ¯8E44    ¯9E44    ¯1E45
¯1.1E45  ¯1.2E45  ¯1.3E45  ¯1.4E45
¯1.5E45  ¯1.6E45  ¯1.7E45  ¯1.8E45
¯1.9E45  ¯2E45    ¯2.1E45  ¯2.2E45
```

Figure 3.1. ***X*** **and** ***Y*** **Data Sets.**

The left argument to PUTDATA is the variable name; the right argument is the file name or path. Note that after transferring any data set to DOS the data is not erased and remains available in the workspace.

When terminating APL, the user is placed in the DOS environment. Figure 3.2 shows that the data set transfer was successfully completed. Note that the first line of the data set gives the dimension or "shape" of the data.

By going back to APL environment and typing the commands:

```
X←GETDATA 'X.DAT'

Y←GETDATA 'Y.DAT'
```

the data is transferred correctly from the DOS environment to the APL workspace. Figure 3.3 shows that the data has the same original shape. The data are read into the workspace and assigned to the variables X and Y. The data still remain available as a file in the DOS environment.

```
      )OFF                                  (terminate APL)

C:\APL>TYPE X.DAT                           (type of contents of files)

          5          5
  0.925   1.85   2.775  3.7
  4.625   5.55   6.475  7.4
  8.325   9.25  10.175 11.1
 12.025  12.95  13.875 14.8
 15.725  16.65  17.575 18.5
 19.425  20.35  21.275 22.2
 23.125
C:\APL>TYPE Y.DAT

          2          4          4
  1E60      2E60      3E60      4E60
  5E60      6E60      7E60      8E60
  9E60      1E61     -1E44     -2E44
 -3E44     -4E44     -5E44     -6E44
 -7E44     -8E44     -9E44     -1E45
 -1.1E45  -1.2E45   -1.3E45   -1.4E45
 -1.5E45  -1.6E45   -1.7E45   -1.8E45
 -1.9E45  -2E45    -2.1E45   -2.2E45
C:\APL>
```

Figure 3.2. ***X* And *Y* Data Sets In DOS Environment.**

```
      ρX
5 5
      X
  0.925   1.85    2.775  3.7     4.625
  5.55    6.475   7.4    8.325   9.25
 10.175  11.1    12.025 12.95   13.875
 14.8    15.725  16.65  17.575  18.5
 19.425  20.35   21.275 22.2    23.125


      ρY
2 4 4
      Y
  1E60      2E60      3E60      4E60
  5E60      6E60      7E60      8E60
  9E60      1E61     ¯1E44     ¯2E44
 ¯3E44     ¯4E44     ¯5E44     ¯6E44

 ¯7E44     ¯8E44     ¯9E44     ¯1E45
 ¯1.1E45   ¯1.2E45   ¯1.3E45   ¯1.4E45
 ¯1.5E45   ¯1.6E45   ¯1.7E45   ¯1.8E45
 ¯1.9E45   ¯2E45     ¯2.1E45   ¯2.2E45
```

Figure 3.3. **Getting *X* And *Y* Into APL Workspace.**

B. EC3400 WORKSPACE

EC 3400 is a course at the Naval Postgraduate School that introduces first principles of digital signal processing. The topics covered include: the Discrete

Fourier Transform and the FFT algorithm, flow-graph and matrix representation of filters, ideal filters and approximation, and design of recursive and nonrecursive digital filters, fast convolution and correlation.

The following workspace enables the student to solve computer assignments on the above subjects in a simple way. This workspace contains functions such as: linear convolution, circular convolution, FFT, inverse FFT, and complex arithmetic. Some of those functions will be described later. In addition, the EC3400 workspace contains functions which allows the student to design one-dimensional filters. The methods used are the Fourier method and the frequency sampling method and windowing of the filters coefficients. The full description of the workspace is given in Appendix B. The following is an example of how to use this workspace.

EC 3400 Computer Assignment [4]
A highpass filter is to have an analog cutoff frequency of 6kHz. The filter is to be implemented by a digital filter having a sampling frequency of 40kHz. Plot the frequency response of the filter with and without a Hamming window if 51 coefficients are used.

Solution

The solution given here uses functions provided in the EC3400 workspace. In an actual classroom assignment, students may be asked to generate their own functions or to solve the problem in their own way. By using the function DIGFREQ, the digital cutoff frequency is calculated. (Figure 3.4). This function, like several others is included for convenience only; the calculation in converting radian to digital frequency can be done easily in the workspace by using the built-in arithmetic operations. Figure 3.5 describes the ideal frequency response to be generated.

By using the function HPCOEFF, 51 causal impulse response coefficients are calculated. (See Figures 3.6 and 3.7). The function HPCOEFF uses the following equation:

$$h_{LP}(n) = k/\pi(n-I)\sin([n-I]\theta_c), \quad n = 0, 1, 2, \ldots, 2I$$

```
      TC←40000 DIGFREQ 8000
      TC
1.256637061

      TC÷PI
0.4
```

Figure 3.4. Calculating a Digital Cutoff Frequency.

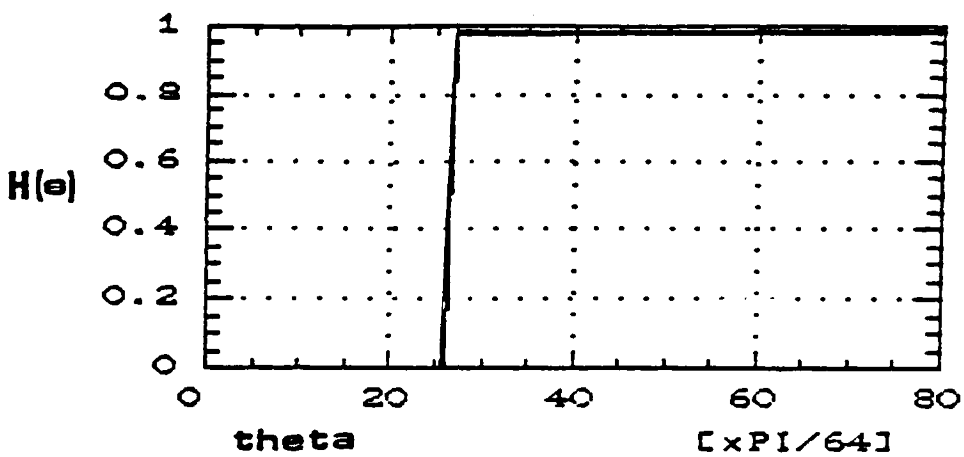


Figure 3.5. The Ideal Frequency Response.

```
      hHP←51 HPCOEFF TC

      hHP
6.863316738E¯17 ¯0.01261377881 ¯8.134689424E¯3 8.504448034E¯3 0.01441574
      2.339861191E¯17 ¯0.01593319429 ¯0.01039432538 0.01100575628
      0.01892066822 2.339861191E¯17 ¯0.02162362082 ¯0.01439214283
      0.01559148806 0.02752097195 2.339861191E¯17 ¯0.0336367435 ¯0.02338
      0.02672826525 0.05045511524 2.339861191E¯17 ¯0.07568267286
      ¯0.06236595225 0.09354892838 0.3027306915 ¯0.6 0.3027306915
      0.09354892838 ¯0.06236595225 ¯0.07568267286 2.339861191E¯17
      0.05045511524 0.02672826525 ¯0.02338723209 ¯0.0336367435 2.3398611
      0.02752097195 0.01559148806 ¯0.01439214283 ¯0.02162362082
      2.339861191E¯17 0.01892066822 0.01100575628 ¯0.01039432538
      ¯0.01593319429 2.339861191E¯17 0.01441574721 8.504448034E¯3
      ¯8.134689424E¯3 ¯0.01261377881 6.863316738E¯17
```

Figure 3.6. Calculating the Impulse Response Coefficients.

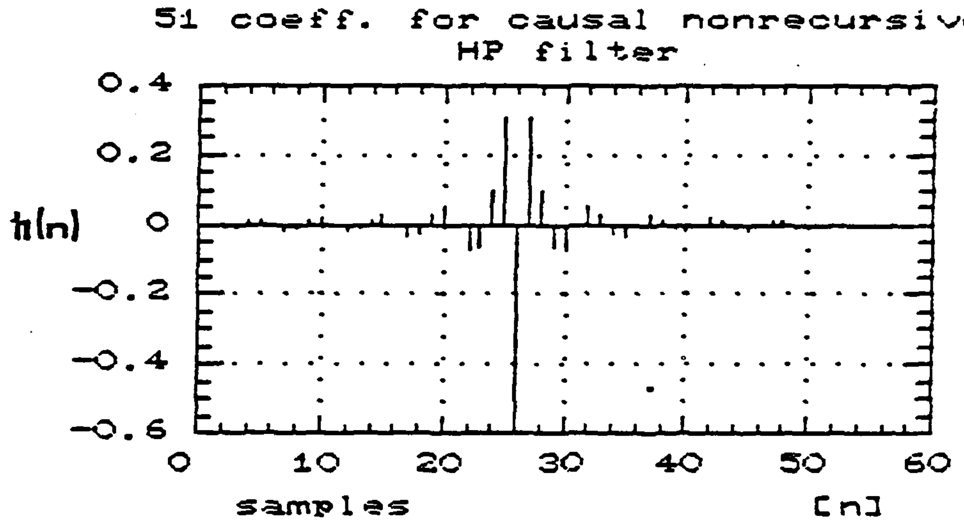


Figure 3.7. The Impulse Response Coefficients.

The function HAMMING generates "I" samples of a shifted Hamming window using the following equation:

$$W(n) = 0.54 + 0.46\cos\left(\frac{2\pi n}{I-1}\right), 0 \leq n \leq I-1$$

This is multiplied by the filter coefficients to obtain the windowed coefficients. (See Figures 3.8 and 3.9).

By using the function FREQRES, the frequency response with and without a window is generated. This is done by calculating the DFT of the filter coefficients

```
      W←HAMMING 51

      W
0.08087246878 0.08783237415 0.1016466798 0.122105975 0.1489001176 0.1816
      0.2197783849 0.2627880673 0.31 0.3606984983 0.4141150246 0.4694398
      0.5258342731 0.5824434454 0.6384092183 0.6928832078 0.7450396437
      0.7940878876 0.8392844181 0.8799441019 0.9154505797 0.9452656094
      0.9689372255 0.9861065906 0.9965134344 1 0.9965134344 0.9861065906
      0.9689372255 0.9452656094 0.9154505797 0.8799441019 0.8392844181
      0.7940878876 0.7450396437 0.6928832078 0.6384092183 0.5824434454
      0.5258342731 0.4694398388 0.4141150246 0.3606984983 0.31 0.2627880
      0.2197783849 0.1816229357 0.1489001176 0.122105975 0.1016466798
      0.08783237415 0.08087246878

      hHPW←hHP×W
```

Figure 3.8. Calculating the Hamming Window Samples.

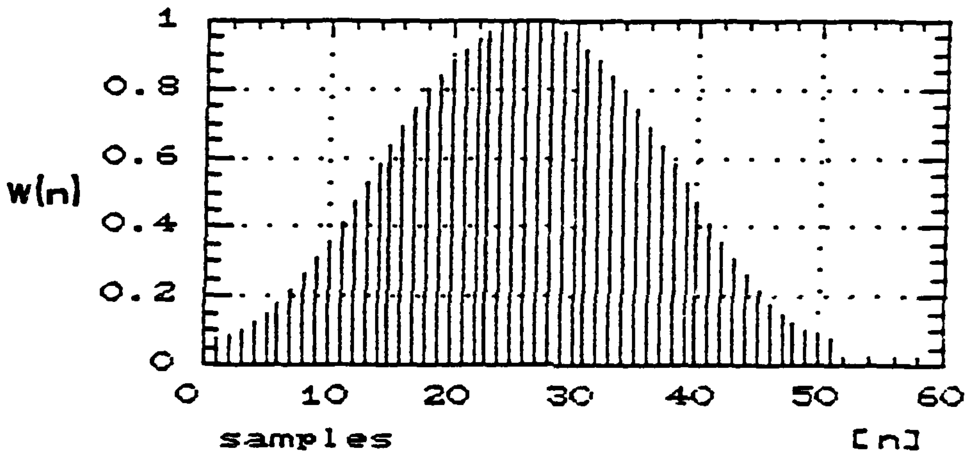


Figure 3.9. The Hamming Window.

and finding the magnitude of the result as shown in a section of the FREQRES function (Figure 3.10). The following describes the APL commands which generate the frequency responses and Figures 3.11 and 3.12 show the corresponding results.

```
HHP←128 FREQRES hHP
HHPW←128 FREQRES hHPW
```

```
      ∇FREQRES[⎕]∇
[0]   HW←M FREQRES h;⎕IO;N;H
[1]   ⍝
[2]   ⍝  GENERATING FREQUENCY RESPONSE MAGNITUDE AND PHASE FROM TIME DOMAIN
[3]   ⍝  COEFFICIENTS
[4]   ⍝
[5]    ⎕IO←0
[6]    N←(M-⍴h)⍴0
[7]    h←h,N
[8]    H←FFT h
[9]    HW←(M÷2)↑(XMAGN H)
[10]   PW←(M÷2)↑(XPHAS H)
[11]   →0
[12] ⍝  THE FUNCTIONS FFT,XMAGN,XPHASE ARE USED.
[13] ⍝  IN ORDER TO RECEIVE A BETTER RESOLUTION THE COEFFICIENTS ARE ZERO-
[14] ⍝  PADDET TO 'M' SAMPLES.
[15] ⍝  M←NUMBER OF SAMPLES INCLUDING ZERO PADDING (USE ONLY RADIX TWO NO.
[16] ⍝  h←THE FILTER COEFFICIENTS [VECTOR]
[17] ⍝  HW:FREQUENCY RESPONSE MAGNITUDE (M÷2 SAMPLES)
[18] ⍝  PW:FREQUENCY RESPONSE PHASE (M÷2 SAMPLES) [RAD.]
```

Figure 3.10. Using the Function FFT.

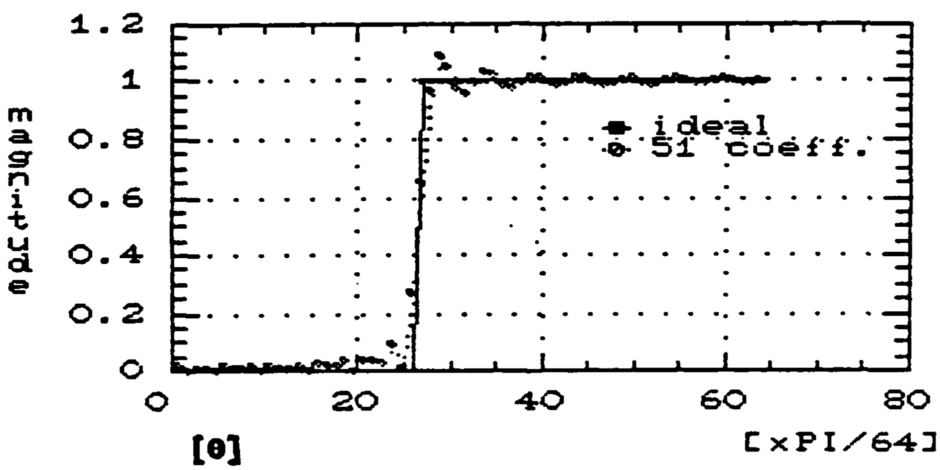


Figure 3.11. The Frequency Response Using 51 Coefficients.

C. EC3410 WORKSPACE

EC3410 is a course that gives an introduction to discrete-time random process. The topics covered are: description of discrete-time random signals and random vectors, linear transformations, sampling of continuous-time random signals, estimation, and spectral analysis. The FC3410 workspace was developed in order to enable the student to solve computer assignments in digital signal processing and

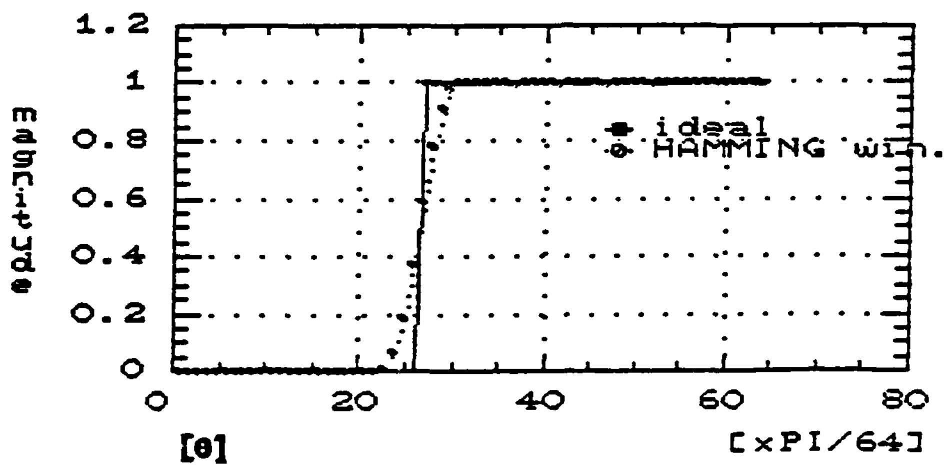


Figure 3.12. The Frequency Response Using Hamming Window.

learn to write algorithms in an efficient way without doing excessive programming. This workspace contains functions that compute the mean, circular convolution, linear convolution, fast Fourier transform, power spectrum and other quantities. A full description of the workspace functions is given in Appendix B.

By using the STATGRAPHICS software package with the EC3410 workspace, a graphic capability is achieved. The following is a computer assignment in statistical dsp and its solution using the EC3410 workspace and the STATGRAPHICS software.

Note that in the actual assignment, students are required to write some of the functions that are used below. However developing the functions in APL using the tools provided is not difficult.

EC 3410 COMPUTER ASSIGNMENT [5]

1. Three data sets SL1.DAT, SL2.DAT, and SL3.DAT are given on a disk. The mean of the signal data SL1.DAT is 0.75. Subtract this constant value from the signal to generate a new zero-mean signal SL1′ that you will use in this assignment. The other data sets (SL2, SL3) have zero mean. Do the remaining parts for each of the data sets SL1′, SL2, and SL3.

2. Using the correlation method, generate a 3 by 3 Toeplitz correlation matrix for the signal data. Print this matrix.

3. Solve a set of Normal equations involving the correlation matrix that was just generated to obtain the 2nd order linear predictive filter parameters and the prediction error variance.

4. Apply the filter to the original data set and generate the prediction error signal. Plot this signal and compute its variance. Does it compare well with the theoretical prediction error variance you obtained by solving the normal equations?

Solution

1. By using the function MEAN, the SL1 data set mean is computed and is subtracted from the original SL1. The new data set SL1′ has a zero-mean. (Figure 3.13)

```
      M←MEAN SL1
      M
2.065552048

      SL1←SL1-M
      M←MEAN SL1
      M
6.982261991E¯17
```

Figure 3.13. Using the Function MEAN.

```
        R1←3 SACF SL1

        R1
8.575461794 8.055195789 7.537100879

        K1←COV R1

        K1
 8.575461794 8.055195789 7.537100879
 8.055195789 8.575461794 8.055195789
 7.537100879 8.055195789 8.575461794
```

Figure 3.14. Using the Functions SACF to Evaluate the Covariance for SL1.DAT.

```
        R2←3 SACF SL2

        R2
1.214143223 0.589008547 0.2762704521

        K2←COV R2

        K2
 1.214143223  0.589008547  0.2762704521
 0.589008547  1.214143223  0.589008547
 0.2762704521 0.589008547  1.214143223
```

```
        R3←3 SACF SL3

        R3
9.410119655 ¯8.931635784 8.467433519

        K3←COV R3

        K3
  9.410119655 ¯8.931635784  8.467433519
 ¯8.931635784  9.410119655 ¯8.931635784
  8.467433519 ¯8.931635784  9.410119655
```

Figure 3.15. Using the Functions SACF and COV for SL2 and SL3.

2. By using the function SACF (sample autocorrelation function), the first three correlation function lags are calculated and then by using the function COV a Toeplitz matrix is generated from the correlation function data. The process described in the above paragraph is repeated on the SL2 and SL3 data sets, as seen in Figures 3.14 and 3.15.

3. The function LPFP (linear prediction filter parameters) solves the Normal equations and it calculates the 2nd order filter coefficients and the prediction error as demonstrated in Figure 3.16. (This could also be done without the

```
a1←2 LPFP SL1                          a2←2 LPFP SL2

a1                                     a2
1 ¯0.9666951989 0.02913175079          1 ¯0.4900717314 0.01020142115

PERR                                   PERR
1.008111643                            0.9283051353

                  a3←2 LPFP SL3

                  a3
                  1 0.9593783731 0.01077400633

                  PERR
                  0.9325296297
```

Figure 3.16. Solving the Normal Equations Using the Function LPFP.

LPFP function by setting up simple linear equations and solving them as described in Chapter II.)

4. Using the LCV (linear convolution) function, the three data sets are convolved with the 2nd order filter parameters that were found in the previous step to provide the prediction error sequence. The prediction errors of the signals are determined by computing the initial value of the autocorrelation function of the prediction error sequence as seen in Figure 3.17. As can be seen, the results are very similar to the prediction error values derived by the analysis of Figure 3.16. Figure 3.18 is a plot of the prediction sequence for each data set.

   Additional examples of using the EC3410 workspace are provided in Appendix D.

D. EC4440 WORKSPACE

EC 4440 is a course at the Naval Postgraduate School in multidimensional digital signal processing. The course deals with the analysis of signals that are functions of the two or more independent variables. The course develops both time/space and frequency domains approaches and highlights the subjects that are different from one-dimensional signal processing. Some of the topics covered

```
      a1←2 LPFP SL1

      a1
1 ¯0.9666951989 0.02913175079

      PERR
1.008111643



      a2←2 LPFP SL2

      a2
1 ¯0.4900717314 0.01020142115

      PERR
0.9283051353



      a3←2 LPFP SL3

      a3
1 0.9593783731 0.01077400633

      PERR
0.9325296297
```

Figure 3.17. Calculating The Prediction Error Using The Function PRER.

are: two-dimensional circular and linear convolution, difference equations, recursively computable systems, Fourier analysis, and methods of 2-D filter design. The EC4440 workspace enables the student to solve computer assignments in multidimensional signal processing by using functions such as: 2-D convolution, 2-D Fast Fourier Transform (FFT) and inverse 2-D FFT, and filter design using the windowing method or the McClellan transformation. The plots are provided by using, as before, the STATGRAPHICS software package. More detailed lists and descriptions of the workspace functions are provided in Appendices B (DSP library) and C (functions listings and samples of runs).

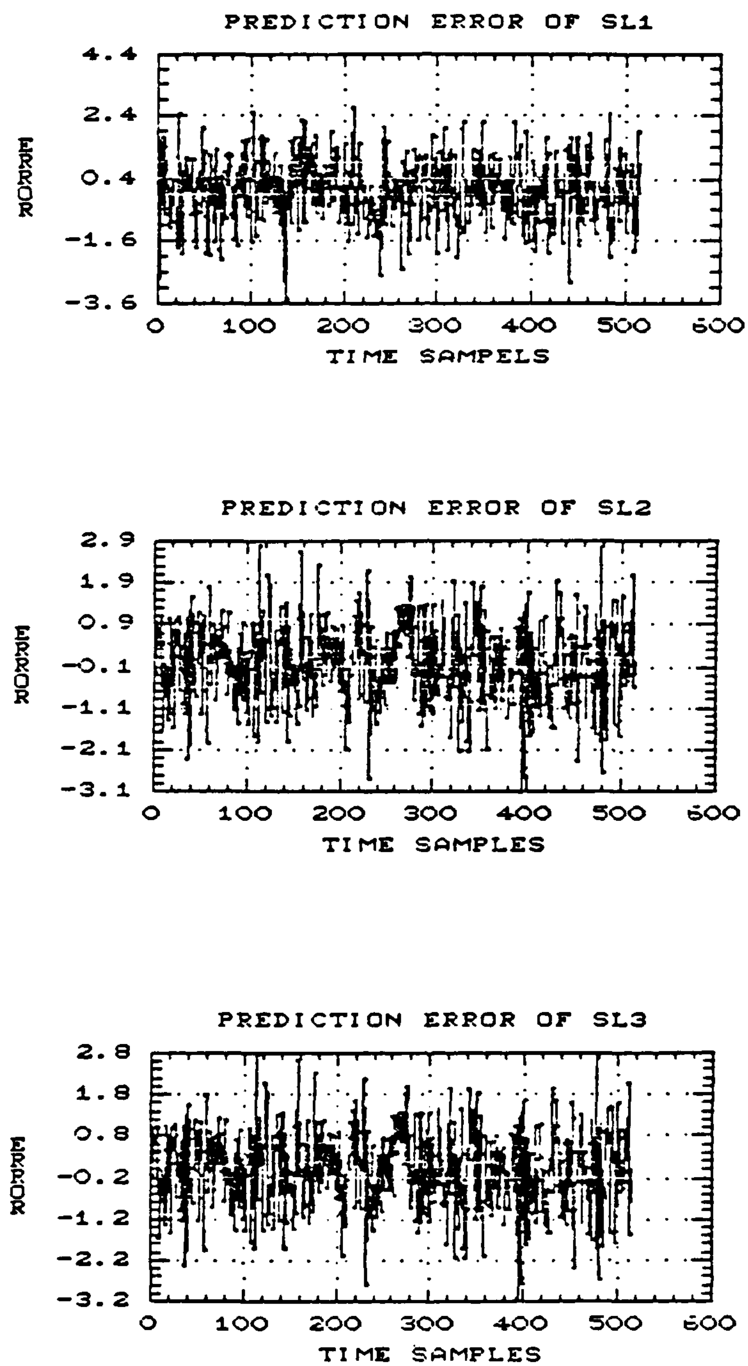
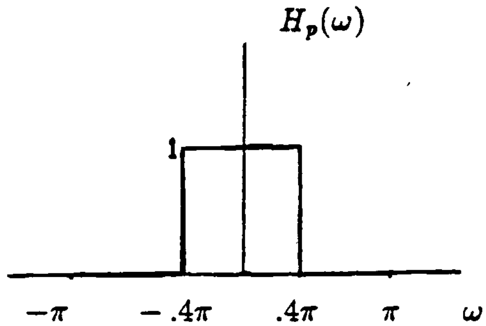


Figure 3.18. The Prediction Errors For SL1, SL2, And SL3.

The following is an example of how to design a lowpass filter using the workspace functions. The design here is done by the McClellan transformation method. Again, as in other examples, the students are asked to write some of these functions.

COMPUTER ASSIGNMENT ON McCLELLAN TRANSFORMATION [6]

Consider the prototype filter:



1. Using a 32 point FFT, find the impulse response $h_p(n)$ and the coefficients $a(n)$ in the representation

$$H_p(\omega) = \sum_{n=0}^{N} a(n) T_n[\cos\omega] \qquad N = 15$$

2. Now let $F(\omega_1, \omega_2) = \frac{1}{2}(-1 + \cos\omega_1 + \cos\omega_2 + \cos\omega_1 \cdot \cos\omega_2)$

   Compute $H(\omega_1, \omega_2) = \sum_{n=0}^{N} a(n) T_n[F(\omega_1, \omega_2)]$ and generate both 3-D and contour plots of this frequency response.

   Note: You can take advantage of the recursion

$$T_n(x) = 2x T_{n-1}(x) - T_{n-2}(x)$$

   and the built-in recursive nature of APL functions to generate the Chebyshev polynomials.

Solution

The function PROFILT is used to specify a 32-sample ideal lowpass prototype filter; then by using the function COEFF the impulse response coefficients are calculated. As shown in the following steps the COEFF function uses the IFFT function to calculate the impulse response, $h(n)$, of the prototype filter.

```
      ∇COEFF[⎕]∇
[0]   a←COEFF Hp;⎕IO;SIZE;HH;H
[1]   ⍝
[2]   ⍝  COMPUTING THE COEFFICIENTS FROM THE PROTOTYPE FILTER (USING IFFT)
[3]   ⍝
[4]    ⎕IO←0
[5]    SIZE←(⍴Hp)÷2
[6]    a←SIZE⍴0
[7]    H←IFFT Hp
```

The filter coefficients are defined as:

$$a(n) \triangleq \begin{cases} h(0), & n = 0 \\ \\ 2h(n), & n > 0 \end{cases}$$

and are found by performing the following steps in the function COEFF.

```
[8]   HH←(1,SIZE)↑H
[9]   HH←SIZE⍴HH
[10]  a[⍳SIZE]←2×HH[⍳SIZE]
[11]  a[0]←a[0]÷2
[12]  →0
[13] ⍝
[14] ⍝  Hp←THE PROTOTYPE SAMPLES
[15] ⍝  a:THE IMPULSE RESPONSE COEFFICIENTS
[16] ⍝  THE IFFT FUNCTION HAS BEEN USED.
[17] ⍝
[18] ⍝  Y.KATZIR.I.A.F., OCTOBER 1987
[19] ⍝
```

```
      HP←PROTFILT 32

      a←COEFF HP

      HP
1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1

      a
0.40625 0.610186859 0.1779849416 ¯0.1365886089 ¯0.1508883476 0.0129955742
      0.110335429 0.0464416897 ¯0.0625 ¯0.07130573562 0.01466457095
      0.07052675554 0.02588834765 ¯0.0504871114 ¯0.05298494156 0.01823057754
```

Figure 3.19. Calculating the Impulse Response Coefficients.

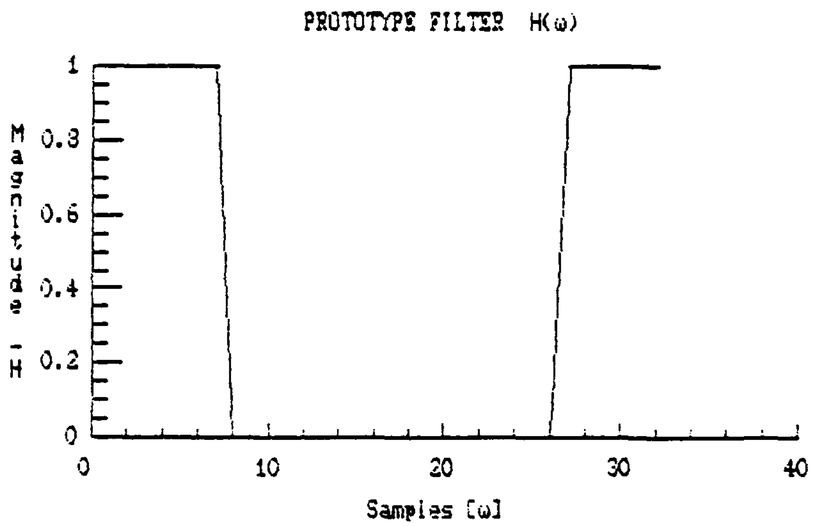


Figure 3.20. 1-D Prototype Filter.

The above calculations are demonstrated in Figure 3.19. (Notice that the prototype filter response is defined on the interval 0 to 2π rather than $-\pi$ to $+\pi$.) Figures 3.20 and 3.21 are plots of the prototype filter and the prototype filter coefficients.

The ideal filter would have a step discontinuance at the points $\omega = 0.4\pi$ and 1.6π but this appears in the plot as slanted lines because the plot uses finite numbers of samples.

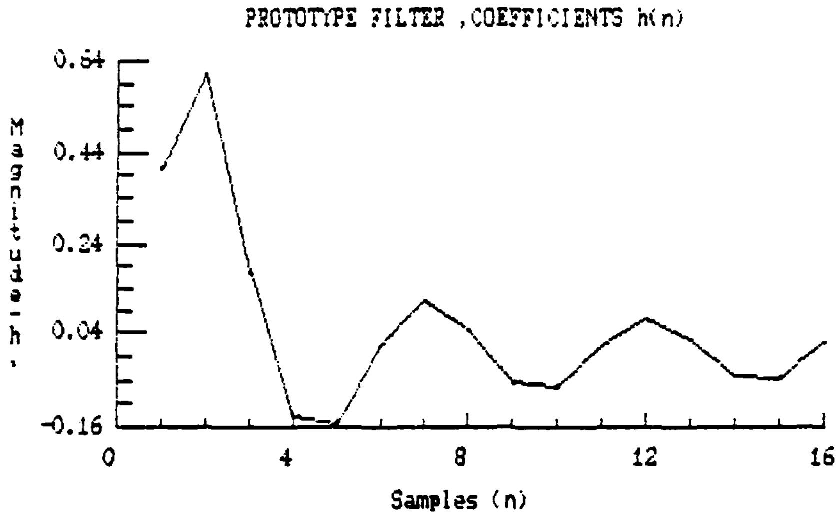


Figure 3.21. The Prototype Filter Coefficients.

The function TRANSFNC generates the transformation function by implementing the following equation:

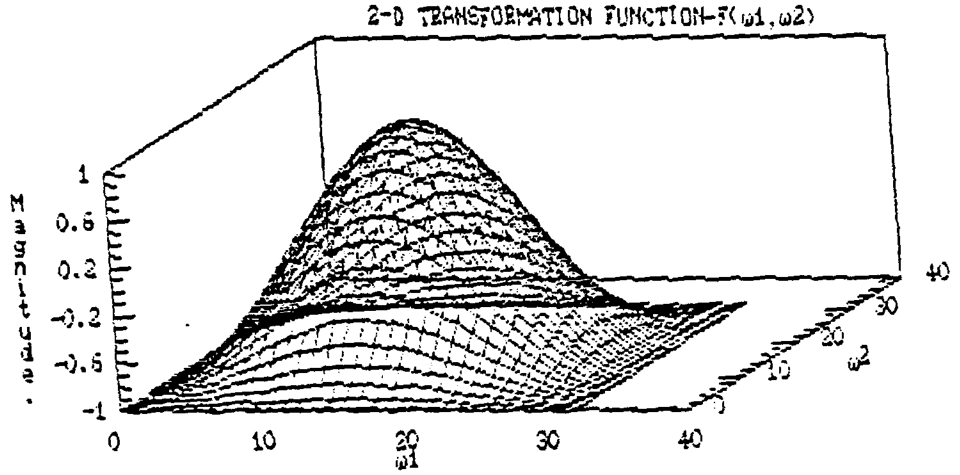
$$F(w1, w2) = 0.5(-1 + \cos\ w1 + \cos\ w2 + \cos\ w1 \cdot \cos\ w2).$$

The surface plot capability of STATGRAPHICS is used to form a 3-D plot and a contour plot of the transformation function as seen in Figure 3.22.

By using the function MCCLEL the frequency response of the desired filter is generated. The function generates the nth Chebyshev polynomial, $Tn[F(w1, w2)]$ by using the recursion $T_n(x) = 2(x)T_{n-1}(x) - T_{n-2}(x)$. By using the following summation:

$$\sum_{n=0}^{N} a(n)T_n\left[F(w1, w2)\right]$$

FW+TRANSFNC 32



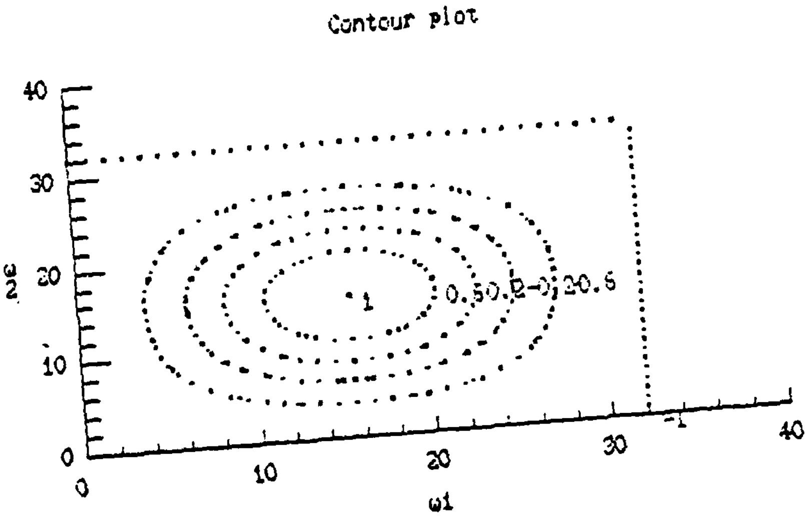


Figure 3.22. The Transformation Function.

the frequency response $H(w1, w2)$ is achieved. The above steps are shown in a section of the MCCLEL function.

```
    ∇MCCLEL[⎕]∇
  Hww←a MCCLEL F;⎕IO;N;INTER;NUMBER;INTER1;INTER0
  ⍝
  ⍝  DESIGNING 2-D FILTER USING MEcCLELLAN TRANSFORMATION
  ⍝
   ⎕IO←0
   INTER0←INTER1←INTER←Hww←(⍴F)⍴0
   NUMBER←⍴a
   N←0
  LOOP:→((N=0),N=1)/A,B
   INTER←(2×F×INTER1)-INTER0
   →C
  A:INTER←(⍴F)⍴1
   INTER1←INTER
   →C
  B:INTER←F
  C:⎕←'CALCULATED CHEB. POLYNOM OF ORDER:',⍕N
   Hww←Hww+(a[N]×INTER)
   INTER0←INTER1
   INTER1←INTER
   →(NUMBER>N←N+1)/LOOP
   →0
  ⍝
  ⍝  F←THE TRANSFORMATION MATRIX (CALCULATED USING TRANSFORMATION
  ⍝  FUNCTION)
  ⍝  a←THE PROTOTYPE IMPULSE RESPONSE COEFFICIENTS
  ⍝  Hww:THE 2-D FREQUENCY RESPONSE.
  ⍝
  ⍝  Y.KATZIR.I.A.F., OCTOBER 1987
  ⍝
```

Figure 3.23 shows the frequency response plot of the lowpass filter. In order to apply a status report during calculation, a status printout was added to the function as seen in Figure 3.24.

E. DISCUSSION

The workspaces and functions developed in this thesis provide a complete set of tools for solving the problems currently assigned to students in courses EC 3400, EC 3410, and EC 4400. They also provide capabilities to solve many other problems in DSP.

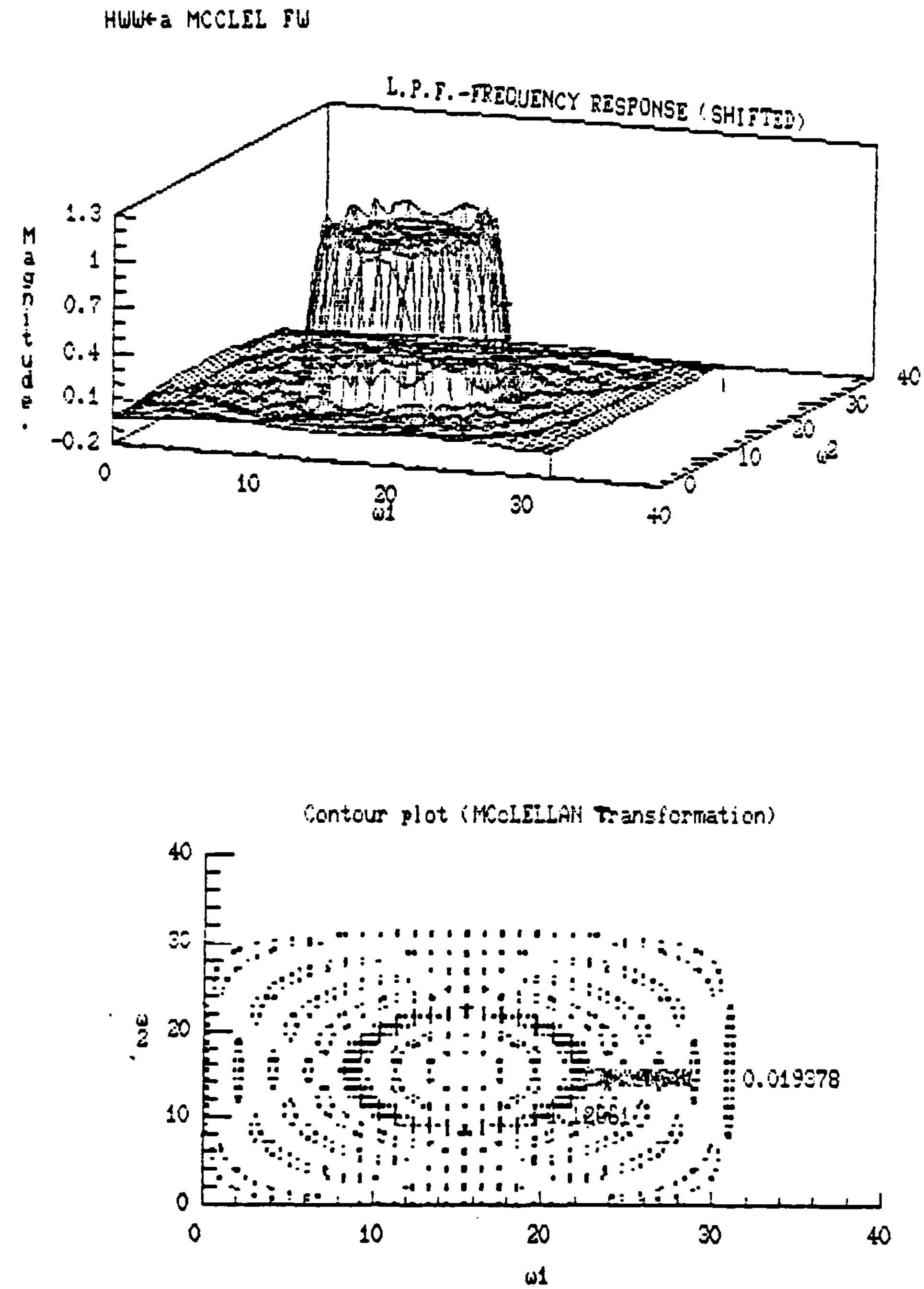


Figure 3.23. The Frequency Response of the Lowpass Filter.

```
CALCULATED CHEB. POLYNOM OF ORDER:0
CALCULATED CHEB. POLYNOM OF ORDER:1
CALCULATED CHEB. POLYNOM OF ORDER:2
CALCULATED CHEB. POLYNOM OF ORDER:3
CALCULATED CHEB. POLYNOM OF ORDER:4
CALCULATED CHEB. POLYNOM OF ORDER:5
CALCULATED CHEB. POLYNOM OF ORDER:6
CALCULATED CHEB. POLYNOM OF ORDER:7
CALCULATED CHEB. POLYNOM OF ORDER:8
CALCULATED CHEB. POLYNOM OF ORDER:9
CALCULATED CHEB. POLYNOM OF ORDER:10
CALCULATED CHEB. POLYNOM OF ORDER:11
CALCULATED CHEB. POLYNOM OF ORDER:12
CALCULATED CHEB. POLYNOM OF ORDER:13
CALCULATED CHEB. POLYNOM OF ORDER:14
CALCULATED CHEB. POLYNOM OF ORDER:15
```

Figure 3.24. Status Printout During Calculation of Chebyshev Polynomial.

Some of the functions developed and included in the workspace are finite specialized and were specifically coded as examples of solving the particular DSP homework assignments. Others are provided as tools that can be applied to many different problems. As already mentioned, it is very easy to expand this set of tools to include others.

In using the workspaces and functions, the instructor will usually provide only a subset of the functions developed here to students enrolled in the course. Students would then use the functions provided, as well as the general features of APL to solve the assigned problems. Since the workspaces are extremely modular, an instructor will find it easy to select only those functions that he wants students to have and provide others at a later time.

IV. SUMMARY AND CONCLUSION

The purpose of this thesis was to develop a convenient set of software tools for students and instructors in order to solve computer assignments in digital signal processing on personal computers using APL.

The main requirement was achieved by developing APL workspaces with sets of functions that could be applied to perform DSP operations. Computer assignments in EC 3400. EC 3410, and EC 4440 are easily solved by using this software package. The user needs to have basic knowledge in APL and needs to know how to use part of the STATGRAPHICS package in the APL environment. However, this basic knowledge is easily acquired.

The structure of the APL software is very flexible and convenient to use. By combining a few functions, the user can achieve complicated calculations with little programming effort. Any workspace can be expanded by the student or professor by adding new functions.

The results of this thesis demonstrate that the APL is easy and convenient to use, especially in solving digital signal processing problems. This thesis will hopefully make it easier for new students to start using the language.

APPENDIX A
USER MANUAL

A. SYSTEM REQUIREMENTS

The software package contains the following disks:

1. APL*PLUS Executive Program version 5.1 or later by STSC
2. STATGRAPHICS - Statistical Graphics System, version 2.1 or later by STSC
3. Utility workspace, and EC3410, EC4440, and EC3400 workspaces

STATGRAPHICS requires the following hardware and system software:

1. IBM PC, PC-XT, PC-AT or compatible
2. 512K RAM
3. Keyboard with APL characters
4. Two double-sided, double-density disk drives or one floppy disk and hard disk.
5. Graphics adapter card
6. DOS, version 2.0 or later
7. Dot matrix graphics printer (optional)
8. Math coprocessor (optional but recommended)

The software package can be operated from floppy disks or from the hard disk.

B. PREREQUISITES FOR USER

In order to use the package efficiently, the user should have a basic knowledge of APL.

In addition, it is strongly recommended that a copy of the STATGRAPHICS manual [3] be available to the user since this Appendix refers the user to Ref. 3

for more explanations. The user should not have loaded any special DOS resident programs since this may interfere with the proper execution of STATGRAPHICS and cause the system to "hang up."

C. HOW TO GET INTO THE DSP LIBRARY

Follow the instructions below to start the session:

1. Start the computer using DOS.

2. If you are using a dual floppy disk drive, place the APL*PLUS executive disk in the default disk drive. If you are using a hard disk system, make sure that the directory with the APL and STATGRAPHICS program (usually the APL directory) is your current directory.

Enter: APLCOM if you are using APL version 5.1 or
Enter: A if you are using APL version 6.4.

These commands call up batch files specifically for each version. A batch file contains a set of commands that must be executed before it runs the APL program in order to have the software matched with the specific hardware.

The system will respond as follows:

```
APL*PLUS PC APPLICATION DEVELOPMENT SYSTEM

Version 5.1   Serial Number 134257    Copyright 1985, STSC, Inc.
All rights reserved.  Unauthorized reproduction of this software
is prohibited and violates U.S. Copyright Laws.  APL*PLUS is a
registered trademark and service mark of STSC, Inc.

CLEAR WS  11/20/1987 14:31:54
```

3. Place the desired DSP library disk in the default disk drive, as for example, the disk containing the UTILITY workspace.

Enter:)LOAD UTILITY

If you are using a hard disk and want to load the workspace from drive A, enter:)LOAD 0 UTILITY. The "0" (zero) stands for drive A, "1" stands for

drive B, and "2" stands for drive C (hard disk). The system will respond as follows:

```
         )LOAD 0 UTILITY
0 UTILITY SAVED 12/13/1987 15:12:28
                  -------------------------------------------------------------
                  | NAVAL POSTGRADUATE SCHOOL              MONTEREY.CALIFORNIA |
                  |-----------------------------------------------------------|
                  |                    UTILITIES WORKSPACE                     |
                  |                                                            |
                  |                         THESIS BY                          |
                  |                      Y.KATZIR.I.A.F.                       |
                  |              ADVISOR:PROF. C.W. THERRIEN.                  |
                  | VERSION 1.0                                 SEPTEMBER 1987 |
                  -------------------------------------------------------------
```

At this point you can use the UTILITY workspace as described in Appendices B, C, and D.

4. Before terminating the session save your results by entering:

)SAVE UTILITY or)SAVE 0 WSID (WSID means your file name)

   the system will respond:

```
        )SAVE
0 UTILITY SAVED 11/19/1987 00:  24:06
```

5. To terminate APL enter:)OFF and you will be automatically returned to DOS.

D. USING STATGRAPHICS WITH APL

Follow the instructions below to start the session:

1. Start the computer under DOS. If a graphics printer is attached be sure that the printer is set up for graphics printing. (DOS has to run graphics.com when system boots.)

2. Load APL as described in the previous section (C.b.)

3. After the APL*PLUS PC system is loaded, replace the APL disk with the STATGRAPHICS start-up disk and enter:

)LOAD STATGRAF

> **WARNING:** When using the STATGRAF workspace use only the)PCOPY command to enter functions or variables into the workspace.

During the loading procedure, the system will display a copyright banner and the following message:

''SYSTEM INITIALIZING. PLEASE BE PATIENT. THIS WILL TAKE A FEW MOMENTS.''

When initialization is complete, the system will prompt for questions that are related to the computer configuration. Select the appropriate answers (See Ref. 3, pg. 3-1 to 3-3 for more details.) The system will ask if you want to save the settings previously entered in order to have an automatic logon the next time you start STATGRAPHICS. The Main Menu will then be displayed as in Figure A.1.

```
                    STATGRAPHICS Statistical Graphics System

DATA MANAGEMENT AND SYSTEM UTILITIES        TIME SERIES PROCEDURES
  A. Data Management                          L. Forecasting
  B. System Environment                       M. Quality Control
  C. Report Writer and Graphics Replay        N. Smoothing
  D. Plotter Interface                        O. Time Series Analysis

PLOTTING AND DESCRIPTIVE STATISTICS         ADVANCED PROCEDURES
  E. Plotting Functions                       P. Categorical Data Analysis
  F. Descriptive Methods                      Q. Multivariate Methods
  G. Estimation and Testing                   R. Nonparametric Methods
  H. Distribution Functions                   S. Sampling
  I. Exploratory Data Analysis                T. Experimental Design

ANOVA AND REGRESSION ANALYSIS               MATHEMATICAL AND USER PROCEDURES
  J. Analysis of Variance                     U. Mathematical Functions
  K. Regression Analysis                      V. Supplementary Operations

        Use cursor keys to highlight desired section.  Then press ENTER.
 1Help    2Edit    3Savscr 4Prtscr 5       6Go      7Vars    8Cmd      9Review 10Quit
INPUT     11/18/87  13:08 STATGRAPHICS Vers. 2.1
```

Figure A.1. Statgraphics Main Menu.

4. When you are in the Main Menu, press the ESC key to enter APL immediate execution mode. Execute the function. Enter: RESTART to return to the menu-control mode.

Since the complete set of functions in the STATGRAPHICS system exceeds the memory space available on most personal computers, you should load only those functions and variables that are necessary to perform the desired tasks. For DSP applications the major procedures used are described in the following section.

D. STATGRAPHICS MAJOR PROCEDURES USED

After starting STATGRAPHICS, the first display is the Main Menu (See Figure A.1.)

The Main Menu contains 22 sections under six major categories.

The Main Menu gives you access to all the data analysis and graphics procedures.

A highlighted cursor bar is displayed on the Main Menu screen. In order to select any desired Menu Procedure, move the cursor using the arrow keys and press: ENTER. STATGRAPHICS procedures are stored in four disks. When using floppy disks carefully follow the instruction on the screen.

The following is a short description of the most frequently used procedures in this DSP software package:

1. System Environments [3]

The STATGRAPHICS system environment section is under category:

``DATA MANAGEMENT AND SYSTEM UTILITIES''

This section allows you to modify the STATGRAPHICS system environment to fit your needs. This section has nine procedures. Those most often used are:

```
                          Screen Options
=====================================================================

        Normal Text                          Graphics Mode
        ===========                          =============
Foreground Color (0-7):     7    * Resolution (1=Low,2=High):   2
Background Color (0-7):     1      Background Color (0-15):     0
Border Color (0-15):        1      Palette Number (0-1):        1
Intense (0-1):              0
Blinking (0-1):             0    * not supported on all boards

        Active Text                        Highlighted Text
        ===========                        ================

Foreground Color (0-7):     0      Foreground Color (0-7):      0
Background Color (0-7):     5      Background Color (0-7):      2
Intense (0-1):              0      Intense (0-1):               0
Blinking (0-1):             0      Blinking (0-1):              0
```

Figure A.2. The Screen Options Menu.

1. Screen option: Allows you to change colors, text, and properties of the graphics screen. (See Figure A.2).

2. Graphic options: Allows you to change sizes, orientation, and positions of the graphics screen. (See Figure A.3.)

```
                          Graphics Options
=====================================================================
                  Line colors:  123456789ABCDEF12345
                  Point colors: 56789ABCDEF123451888
                  Paint colors: 89ABCDEF123451234567
                  Text color:   1

Axis  Color  Ticmarks Text Size                  Color     Viewpoint
----  -----  -------- ---- ----                  -----     ---------
 X      1       In     1    1      Frame:  Yes             X    600
 Y      1       In     1    1      Grid:   Both    1       Y  -1600
 Z      1       In     1    1      Border:  None   1       Z    800
Title1                 1    1
Title2                 1    1     Graphics Cursor: Yes  Graphics: Yes
                                  Pointsize: .005 Ticmark length: .040
 Window      Origin    Width      Split: 1        Position: 1
 ------      ------    -----
Horizontal    .200      .750        Printing Zoom Factors
Vertical      .200      .640      Vertical: 1        Horizontal: 1
                                  Page Orientation: Vertical
```

Figure A.3. Graphics Options Menu.

2. <u>Plotting Functions</u> [3]

Plotting functions are found on the Main Menu under the category:

``PLOTTING AND DESCRIPTIVE STATISTICS''

STATGRAPHICS provides seven plotting procedures you can use to produce a variety of plots. The following is an example of using the X-Y Time Plot procedure.

1. Start in the APL environment and define the variables X and Y as follows:

```
0 STATGRAF                      APL                    Ins
You are now under control of the APL interpreter.
To resume menu control, type:
      RESTART
      X←ι10
      X
1 2 3 4 5 6 7 8 9 10
      Y←1 2 2.5 3.3 4 5 6 7.5 8.8 9.5
      Y
1 2 2.5 3.3 4 5 6 7.5 8.8 9.5
```

Enter: RESTART

Select the X-Y Line and Scatterplots procedures from the Main Menu.

2. Fill in the Menu entry as demonstrated in Figure A.4.

(USE CAPITAL LETTERS)

```
                         X-Y Line and Scatterplots
------------------------------------------------------------------------
                                                                    Log
X: X                                                                No
Y: Y                                                                No

Point labels:

Point codes:
Point colors:
Std. err. X:
Std. err. Y:

Points: Yes
Lines:  Yes
```

Figure A.4. ***X-Y* Plotting Menu**

3. Press the F6 key in order to see the plot. (See Figure A.5.)

4. Press the F4 key to print the screen. (if graphics printer is attached)

5. Press the F5 key to make any changes in the plot (including titles).

6. Press the F10 key to leave the screen and to return to the menu.

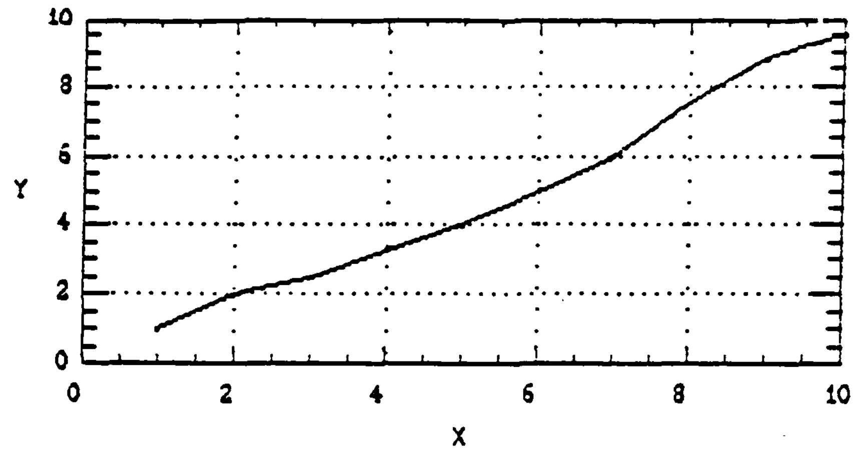


Figure A.5. The Plot of Y vs X.

3. Response Surface Plotting [3]

This procedure produces 3-D surface plots and contour plots, for previously generated two dimensional functions. The procedure is under section: "Experimental Design" on the main menu under

"ADVANCED PROCEDURES''

The following is an example of how to use this procedure:

1. Start with a definition of a matrix of values for a 2-D function in the APL environment as in the following example:

```
——→ XX←16 16⍴⍳256
      XX
   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32
  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48
  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64
  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96
  97  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112
 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128
 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176
 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192
 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208
 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224
 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240
 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256
```

2. Restart STATGRAF and select the section "experimental design." The following will be displayed.

```
EXPERIMENTAL DESIGN

    1.  Full and Fractional Factorials
    2.  Central Composite Designs
    3.  Alias Structure
--> 4.  Response Surface Plotting
```

3. Select "Response Surface Plotting" and make the entry in the menu as described in Figure A.6.

```
                         Response Surface Plotting
--> X-axis minimum: 0                    --> Y-axis minimum: 0
--> X-axis maximum: 16                   --> Y-axis maximum: 16
--> Number of x-axis intervals: 15       --> Number of y-axis intervals: 15
--> Function type: Matrix input
```

Figure A.6. Surface Plotting Menu.

4. Press the F6 key and fill in the entries as shown in the example of Figure A.7.

```
                         Response Surface Plotting
    X-axis minimum: 0                    Y-axis minimum: 0
    X-axis maximum: 16                   Y-axis maximum: 16
    Number of x-axis intervals: 15       Number of y-axis intervals: 15
    Function type: Matrix input

--> Grid matrix: XX
```

Figure A.7. Selecting the Function Definition.

```
                         Surface Plotting Options
   Top title:     THE MATRIX XX (EXAMPLE)
   (2 lines)
   X-Axis Title: X
   Y-Axis Title: Y
   Z-Axis Title: Z

→ Plot type: Surface

  For surface plots:
   X-axis resolution: 100            X-axis skip factor:  1
   Y-axis resolution: 100            Y-axis skip factor:  1
   Lines parallel to: Both axes      Hidden line removal: None

  For contour plot:
   Type: Lines
   Contour levels:
     or
   Number of Divisions: 10  Minimum: 1           Maximum: 256
```

Figure A.8. Surface Plot Entry.

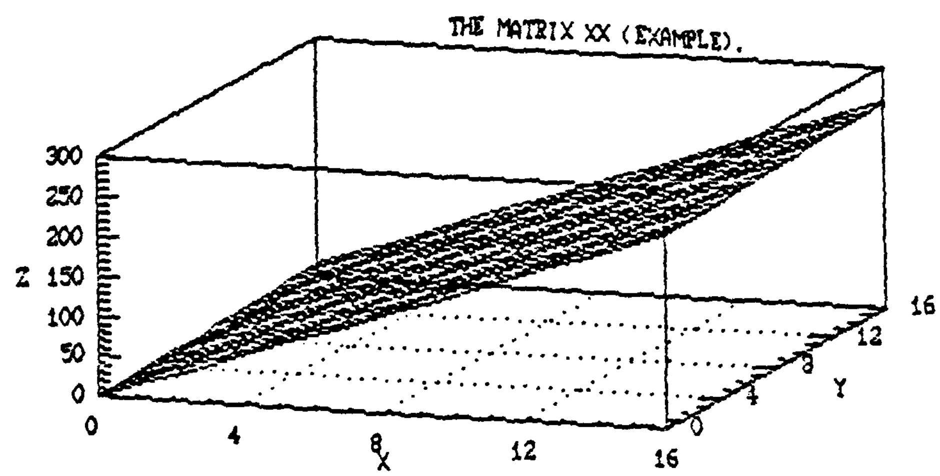


Figure A.9. Surface 2-D Plot.

5. Press the F6 key and enter the values for the surface or contour plot as shown in the example of Figure A.8.

6. Press the F6 key to execute the plot, the F4 key to print the plot, and the F5 key to edit the plot (See Figue A.9).

7. Press the F10 key to leave this screen and return to the menu.

WARNING: Since the number of points in 3-D plots may exceed the memory space available, do not select hidden or partially hidden lines in complicated plots. Select the entry "points" instead of entry "lines" when drawing complicated contour plots.

APPENDIX B
DIGITAL SIGNAL PROCESSING LIBRARY

Listed below are functions' names and brief discussions of the various computer algorithms used to accomplish this thesis research. All programs were written in APL by the author unless otherwise noted. Program listings and samples of runs are given in Appendix C. For each function its name and the page containing its listing in Appendix C are given. Then the calling sequence is given, followed by a brief description.

A. UTILITY WORKSPACE

1. NORD (Pg. 68)

```
Z←N NORD S
```

This function generates a normally distributed random vector using the function UNRD.

The function inputs are:

a. N: Number of elements in the vector

b. S: Vector with two arguments:

*[S1]: The required mean

*[S2]: The required standard deviation

The function's source is the software package GRAFSTAT.

2. UNRD (Pg. 68)

```
R← N UNRD B
```

This function generates a uniformly distributed random vector. The inputs are:

a. N: Numbers of elements in the vector

b. B: Required mean

3. PUTDATA (Pg. 70)

```
F PUTDATA 'A:\WORK\NAME.EXT'
```

This function transfers data (vector, array, etc.) in free format from an APL workspace into an IBM-PC DOS file. The function uses APL NATIVE files [7]. A bell sounds when the data transfer has been completed. The function inputs are:

a. F: The data to be transferred

b. 'A:\WORK\NAME.EXT': The designated native file or path: "A:" Stands for the drive type, "WORK" stands for the directory, and "EXT" stands for extension of the DOS file.

> **WARNING:** The user is not permitted to designate a file that already exists as a DOS NATIVE file.

4. GETDATA (Pg. 72)

```
F←GETDATA 'A:\WORK\NAME.EXT'
```

This function transfers data (vector, array, etc.) from an IBM-PC DOS file to an APL workspace. The function uses APL NATIVE files [7]. A bell sounds when the data transfer has been completed.

'A:\WORK\NAME.EXT': The native file's name or path has to be transferred: "A:" stands for the drive type, "WORK" stands for the directory, and "EXT" stands for extension of the DOS file.

B. EC3400 WORKSPACE [4]

1. DIGFREQ (Pg. 75)

`TC←FS DIGFREQ FC`

This function calculates the digital frequency using the equation:

$$\theta = \omega T = 2\pi f \cdot 1/f_s \tag{B.1}$$

f = The analog frequency

f_s = The sampling frequency

The inputs are:

a. `FS`: The sampling frequency [Hz]

b. `FC`: The analog frequency [Hz]

The output is:

`TC`: The digital frequency [rad.]

2. LPCOEFF (Pg. 76)

`hLP←n LPCOEFF TC`

This function calculates the $2I+1$ lowpass FIR filter coefficients using the Frequency Transformation and the equation:

$$h_{LP}(n) = k/\pi(n-I)\sin([n-I]\theta_c), n = 0, 1, 2, \ldots, 2I \tag{B.2}$$

where $k = 1$ and θ_c = The digital cutoff frequency

The calculated coefficients are for a causal nonrecursive filter. In order to generate noncausal coefficients use the function SHIFT. The inputs are:

a. `n`: Number of required coefficients (Use an odd number.) By choosing a final number of coefficients you automatically define a rectangular window.

b. `TC`: The digital cutoff frequency [rad.]

3. HPCOEFF (Pg. 77)

```
hHP←n HPCOEFF TC
```

This function calculates the $2I + 1$ highpass FIR filter coefficients using the Frequency Transformation method and the equation:

$$h_{HP}(n) = (-1)^{(n-I)} h_{LP}(n), n = 0, 1, 2, \ldots, 2I \qquad (B.3)$$

The calculated coefficients are for a causal nonrecursive (linear phase) filter. In order to generate noncausal coefficients (zero phase filter) use the function SHIFT. The coefficients are normalized. The inputs are:

a. `n`: The number of the required coefficients (Use an odd number.) By choosing a final number of coefficients you automatically define a rectangular window.

b. `TC`: The digital cutoff frequency [rad.]

4. BPCOEFF (Pg. 78)

```
hBP←n BPCOEFF TUL
```

This function calculates the $2I + 1$ bandpass FIR filter coefficients using the Frequency Transformation method and the equations:

$$h_{BP}(n) = [2\cos[(n - I)\theta_o]]\, h_{LP}(n), n = 0, 1, 2, \ldots, 2I \qquad (B.4)$$

$$\theta_c = (\theta_u - \theta_\ell)/2$$

$$\theta_o = (\theta_u + \theta_\ell)/2$$

θ_u = The upper digital frequency

θ_ℓ = The lower digital frequency

θ_c = The digital cutoff frequency

The calculated coefficients are for a causal nonrecursive filter. In order to generate noncausal coefficients use the function SHIFT. The coefficients are normalized. The inputs to the function are:

a. n: The number of the required coefficients (Use an odd number.) By choosing a final number of coefficients you automatically define a rectangular window.

b. TUL: Vector of two arguments

TUL [1]: The upper frequency [rad.]

TUL [2]: The lower frequency [rad.]

5. BSCOEFF (Pg. 79)

```
hBS←n BSCOEFF TUL
```

This function calculates the $2I+1$ bandstop FIR filter coefficients using Frequency Transformation and the equation:

$$\begin{aligned} h_{BS}(0) &= 1 - h_{BP}(0) \\ h_{BS}(n) &= -h_{BP}(n), n = 1, 2, \ldots, 2I \end{aligned} \qquad (B.5)$$

$\theta_c = (\theta_u - \theta_\ell)/2$

$\theta_o = (\theta_u + \theta_\ell)/2$

$\theta_u =$ The upper digital frequency

$\theta_\ell =$ The lower digital frequency

$\theta_c =$ The digital cutoff frequency

The calculated nonrecursive coefficients are for a causal nonrecursive filter. In order to generate noncausal coefficients use the function SHIFT. The coefficients are normalized. The inputs to the function are:

a. n: The number of the required coefficients (Use an odd number.) By choosing a final number of coefficients you automatically define a rectangular window.

b. TUL: Vector of two arguments

TUL [1]: The upper frequency [rad.]

TUL [2]: The lower frequency [rad.]

6. <u>Hamming (Pg. 80)</u>

`W←HAMMING N`

This function generates an $2I+1$ point Hamming window using the equation:

$$W(n) = 0.54 + 0.46\cos\left[\frac{\pi(n-I)}{I}\right], n = 0, 1, 2, \ldots, 2I \qquad (B.6)$$

The window starts at $n = 0$ and has $2I + 1$ samples. The input to the function is

`N`: The number of samples.

7. <u>HANNING (Pg. 81)</u>

`W←HANNING N`

This function generates an $2I+1$ point Hanning (von Hann) window using the equation:

$$W(n) = 0.5 + 0.5\cos\left[\frac{\pi(n-I)}{I}\right], n = 0, 1, 2, \ldots, 2I \qquad (B.7)$$

The window starts at $n = 0$ and has $2I + 1$ samples. The input to the function is:

`N`: The number of samples.

8. <u>BARTLETT (Pg. 82)</u>

`W←BARTLETT N`

This function generates an $2I+1$ point Bartlett (triangular) window using the equation:

$$W(n) = \begin{cases} 1 - \frac{1}{I}(n-I) & ,0 \leq n \leq 2I \\ 0 & \text{, otherwise} \end{cases} \qquad (B.8)$$

The window starts at $n = 0$ and has $2I + 1$ samples. The input to the function is:

N: The number of samples.

9. FREQRES (Pg. 83)

`HW←Z FREQRES h`

This function generates the magnitude and phase of the frequency response from the unit impulse response. The functions: `FFT`, `XMAGN`, `XPHASE` are used. In order to achieve good resolution the coefficients are zero padded to 'Z' samples.

WARNING: The 'Z' needs to be a power of 2.

The inputs to the function are:

a. `Z`: The total number of samples including the zero padding

b. `h`: The impulse response coefficients

The outputs are:

a. `HW`: The frequency response magnitude (Z/2 samples)

b. `PW`: The frequency response response phase (Z/2 samples) [rad.]

10. IDEALF (Pg. 84)

`H←TYPE IDEALF TC`

This function produces the frequency response of various ideal filters. The inputs are:

a. '`TYPE`': The type of the filter to be generated:

'LP': Lowpass

'HP': Highpass

'BP': Bandpass

'BS': Bandstop

b. TC: A vector containing the following:

1. In the case of LP or HP:

   TC [1] = The number of samples

   TC [2] = The cutoff digital frequency

2. In a case of BP or BS:

   TC [1] = The number of samples

   TC [2] = The upper digital frequency

   TC [3] = The lower digital frequency

The frequencies should be normalized to lie in the interval 0 to π. [rad.]

11. FCOEFF (Pg. 86)

`h←n FCOEFF HW`

This function calculates the $2I+1$ point filter coefficients using IFFT (The sampling method.)

The coefficients are for a causal nonrecursive filter. The inputs to the function are:

a. n: The number of the required coefficients (Use an odd number.) By choosing a final number of coefficients you automatically define a rectangular window.

b. HW: The ideal frequency response.

C. EC3410 WORKSPACE [5, 8]

1. SACF (Pg. 88)

`R←LAG SACF DAT`

This function computes the sample autocorrelation function (unbiased) using the following equation:

$$\hat{R}_{yy}^{(k)} = 1/N \sum_{n=0}^{N-1-|k|} Y_{n+|k|} + Y_n, \quad 0 \le k \le N-1 \qquad (B.9)$$

The inputs are:

a. `DAT`: The data block (one-dimensional vector)

b. `LAG`: Number of lags to be calculated (the limit is 300 lags)

2. MEAN (Pg. 90)

`M←MEAN F`

This function calculates the sample mean of a data set using the following equation:

$$m = 1/N \sum_{i=1}^{N} Y^{(i)} \tag{B.10}$$

The input is:

`F`: The data block

3. COV (Pg. 91)

`K←COV R`

This function generates a Toeplitz covariance matrix using covariance function. The input is:

`R`: The covariance function values (vector)

4. LPFP (Pg. 92)

`a←P LPFP DATA`

This function generates the pth order Linear Predictive Filter parameters and calculates the Prediction Error Variance by solving the Normal Equation:

$$\begin{bmatrix} R(0) & R(1) & \dots & R(p-1) \\ R(1) & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ R(p-1) & \dots & & R(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_{p-1} \end{bmatrix} = \begin{bmatrix} \sigma_\epsilon^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{B.11}$$

The inputs are:

a. `P`: The coefficients order

b. `DATA`: The data block

The outputs are:

a. `a`: The filter coefficients

b. `PRER`: The estimate prediction error variance

The following functions are used:

a. `SACF`: Calculates the sample autocorrelation of the data block

b. `COV`: Generates the Toeplitz matrix of the covariance function

5. CC (Pg. 93)

```
Y←x CC h
```

This function performs one-dimensional circular convolution between two inputs ($h(n) \circledast x(n)$). x and h are the arguments to be convolved.

> **WARNING:** x and h need to be the same size and one-dimensional, otherwise an error message will be generated and the convolution will not start.

6. FCC (Pg. 94)

```
Y←x FCC h
```

This function does one-dimensional fast circular convolution between two arguments. The fast operation is established by a shifted matrix multiplication. x and h are the arguments to be convolved. Due to memory size this function is limited to arguments of less than 80 numbers. If the argument is too long a printout message advises the user to use regular circular convolution function.

> **WARNING:** x and h need to be the same size and one-dimensional, otherwise an error message will be generated and the convolution will not start.

7. LCV (Pg. 95)

`Y←x1 LCV h1`

This function does one dimensional linear convolution between two arguments by using the following equation:

$$x_1(n) * h_1(n) = \sum_{m=-\infty}^{\infty} x_1(m)h_1(n-m) \qquad (B.12)$$

$x1$ and $h1$ are the arguments to be convolved.

The function uses the CC function (Circular Convolution) and the linear convolution is produced by zero padding the arguments.

8. FLCV (Pg. 96)

`Y←x1 FLCV h1`

This function does one-dimensional linear convolution (same as LCV function) by using the function FCC in order to achieve a faster computation time.

9. FFT (Pg. 97)

`Z←FFT X`

This function calculates the one-dimensional Discrete Fourier Transform according to the definition

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j(2\pi/N)nk}, k = 0, 1, \ldots, N-1 \qquad (B.13)$$

The input to the function:

`X`: Can be either real vector of length N or a two by N matrix (the 1st row is the real part and the 2nd row is the imaginary part). N must be a power of two.

The output is:

`Z`: A complex two by N matrix (real and imaginary)

The function was written by Professor Paul Penfield, Jr. of M.I.T.

10. IFFT (Pg. 98)

`Z←IFFT X`

This function calculates the one-dimensional Inverse Discrete Fourier Transform according to the definition:

$$x(n) = 1/N \left[\sum_{k=0}^{N-1} X(k) e^{-j(2\pi/N)nk} \right]^{*}, n = 0, 1, \ldots, N-1 \qquad (B.14)$$

The input to the function is:

X: Can be either a real vector of length N or a two by N matrix (the 1st row is the real part and the 2nd row is the imaginary part). N must be a power of two.

The output is:

Z: A complex two by N matrix (real and imaginary)

The function was written by Professor Paul Penfield, Jr. of M.I.T.

11. PSE (Pg. 100)

`INW←ZP PSE DATA`

This function computes the power spectrum estimate (Periodogram) using the equation:

$$\hat{S}_x\left(e^{j\theta}\right) = IN(\theta) = 1/N \left| X\left(e^{j\theta}\right) \right|^2 \qquad (B.15)$$

The function uses the FFT APL function and the inputs are:

a. **DATA**: The data block (one dimensional vector)

b. **ZP**: The length of the zero padding needed for a smooth plot

WARNING: The DATA plus ZP need to be in length which is a power of two

12. PSEB (Pg. 100)

```
INBW←SEG PSEB DATA
```

This function computes the power spectrum estimate with rectangular nonoverlapping window using the equation:

$$
\begin{aligned}
IN'^{(i)}(\theta) &= 1/N \left| X^{(i)} \left(e^{j\theta}\right) \right|^2 \\
B(\theta) &= 1/M \sum_{i=1}^{M} IN'^{(i)}(\theta)
\end{aligned}
\qquad (B.16)
$$

N: The length of the data set

M: The length of a segment

Each segment has N' Points: $N = N' \times M$.

This function uses the FFT and the PSE APL functions and the inputs are:

a. `SEG`: The length of the segments

b. `DATA`: The data block (one-dimensional vector)

The function automatically zero pads the segment up to 256 points.

WARNING: In case of error due to memory size, use fewer segments

13. XSQRT (Pg. 102)

```
XZ←XSQRT XR
```

This function calculates the complex square root of a complex number.

$$XZ = XR^{1/2} \qquad (B.17)$$

14. XTIME (Pg. 102)

```
XZ←XL XTIME XR
```

This function multiplies two complex numbers

$$XZ = XL \times XR \tag{B.18}$$

15. XMAGN (Pg. 103)

```
Z←XMAGN XR
```

This function computes the complex magnitude of a number, the result is a real number

$$Z = |XR| \tag{B.19}$$

16. XPHAS (Pg. 103)

```
Z←XPHAS XR
```

This function computes the phase of the complex number XR, and returns a real number.

$$Z = \measuredangle XR \tag{B.20}$$

17. XCONJ (Pg. 104)

```
XZ←XCONJ XR
```

This function returns the complex congugate of the input.

18. XEPO (Pg. 104)

```
XZ←XEPO XR
```

This function computes the complex exponential

$$XZ = e^{XR} \tag{B.21}$$

19. PI (Pg. 105)

```
Y←PI
```

This function returns the π value.

20. COS (Pg. 105)

```
Y←COS X
```

This function returns the cosine of the input.

21. SIN (Pg. 105)

```
Y←SIN X
```

This function returns the sine of the input.

D. EC4440 WORKSPACE [9]

1. CC2D (Pg. 107)

```
Y←X CC2D H
```

This function calculates two dimensional circular convolution using the equation:

$$Y = h \; ** \; x$$

$$Y(n_1, n_2) = \sum_{n1=0}^{N_1-1} \sum_{n2=0}^{N_2-1} h(n_1, n_2) \; x\left[((n_1 - m_1))_{N1} \, ((n_2 - m_2))_{N_2}\right] \qquad (B.22)$$

Where subscript $N1$ and $N2$ denotes arithmetic modulo $N1$ and $N2$. x and h are the arguments to be convolved.

> **WARNING**: X and H need to be two-dimensional with the same size, otherwise an error message will be generated and the convolution will not start.

2. FFT2D (Pg. 109)

```
Y←FFT2D X
```

This function calculates the two-dimensional DFT using the one- dimensional FFT function. If the input is real, the function converts it to a complex array. If the output array is real valued the second plane is eliminated automatically.

WARNING: The input needs to be a length which is a power of two.

3. IFFT2D (Pg. 110)

```
Y←IFFT2D X
```

This function calculates the two-dimensional Inverse DFT using one-dimensional IFFT function (the same explanation as FFT2D).

4. SHIFT (Pg. 114)

```
Y←SHIFT X
```

This function shifts periodic data to an interval starting at the origin.

5. UNSHIFT (Pg. 114)

```
Y←UNSHIFT X
```

This function shifts periodic data to an interval centered at the origin.

6. SHIFT2D (Pg. 115)

```
Y←SHIFT2D X
```

This function shifts two-dimensional periodic data to an interval starting at the origin

7. UNSHIFT2D (Pg. 115)

```
Y←UNSHIFT2D X
```

This function shifts two-dimensional periodic data to an interval centered at the origin. Each of the following functions generates a two-dimensional window by using a one-dimensional window. Two types of supports are used:

a. A rectangular region of support which is achieved by computing the outer product of two 1-D windows:

$$W_R(n_1, n_2) = W_1(n_1)W_2(n_2) \qquad (B.23)$$

b. A circular region of support which is formed by sampling a circularly rotated, 1-D, continuous function:

$$W_c(n_1, n_2) = W\left(\sqrt{n_1^2 + n_2^2}\right) \qquad (B.24)$$

8. ILPFILT (Pg. 117)

`W←R ILPFILT P`

This function generates a circular based rectangular window using the 1-D equation:

$$W(n) = \begin{cases} 1 & , |n| < R \\ 0 & , \text{otherwise} \end{cases} \qquad (B.25)$$

This can also be used for defining the frequency response of an ideal circular symmetrical lowpass filter. The inputs are:

a. R: The window base radius

b. P: The dimension of the square matrix in which this window is defined.

9. HAMMINGR (Pg. 119)

`WR←II HAMMINGR N`

This function generates a 2-D Hamming rectangular based window using the 1-D equation:

$$W(n) = \begin{cases} 0.54 - 0.46\cos(\pi n/I) & , |n| < I \\ 0 & , \text{otherwise} \end{cases} \qquad (B.26)$$

The inputs are:

a. II: The window base dimension (one-side)

b. N: The dimension of the square matrix in which this window is defined.

10. HAMMINGC (Pg. 121)

```
WC←II HAMMINGC N
```

This function generates a 2-D Hamming window with circular base using equation (B.26). The inputs are:

a. `II`: The window base radius

b. `N`: The dimension of the square matrix in which the window is defined.

11. HANNINGR (Pg. 123)

```
WR←II HANNINGR N
```

This function generates a 2-D Hanning window with rectangular base using the 1-D equation:

$$W(n) = \begin{cases} 0.5\left[1+\cos\left(\pi n/I\right)\right] & , |n| < I \\ 0 & , \text{otherwise} \end{cases} \qquad (B.27)$$

The inputs are:

a. `II`: The window base dimension (one side)

b. `N`: The dimension of the square matrix in which the window is defined.

12. HANNINGC (Pg. 125)

```
WC←II HANNINGC N
```

This function generates a 2-D Hanning window with circular base using equation (B.27). The inputs are:

a. `II`: The window base radius

b. `N`: The dimension of the square matrix which this window is defined.

13. BARTLETTR (Pg. 127)

```
WR←II BARTLETTR N
```

This function generates 2-D Bartlett (triangular) window with rectangular base using the 1-D equation:

$$W(n) = \begin{cases} 1 - \frac{1}{I}n & , 0 \leq n \leq I \\ 1 + \frac{1}{I}n & , -1 \leq n \leq 0 \\ 0 & , \text{otherwise} \end{cases} \qquad (B.28)$$

The inputs are:

a. `II`: The window base radius

b. `N`: The dimension of the square matrix in which the window is defined.

14. BARTLETTC (Pg. 129)

`WC←II BARTLETTC N`

This function generates a 2-D Bartlett window with rectangular base using equation (B.28). The inputs are:

a. `II`: The window base radius

b. `N`: The dimension of the square matrix in which the window is defined.

15. RECTANR (Pg. 131)

`WR←II RECTANR N`

This function generates 2-D rectangular window with rectangular base using equation (B.25). The inputs are:

a. `II`: The window base dimension (one side)

b. `N`: The dimension of the square matrix in which the window is defined.

16. PROTFILT (Pg. 133)

`HPROT←PROTFILT SIZE`

This function produces a frequency response 1-D lowpass prototype filter in order to design a 2-D FIR filter using a transformation. The function SHIFT has been used. The input is:

SIZE: The size of the prototype filter (number of samples).

17. COEFF (Pg. 135)

A←COEFF HP

This function finds the coefficients $a(n)$ in the representation:

$$H(\omega) = \sum_{n=0}^{N} a(n)\cos(\omega n)$$

where:

$$a(n) \triangleq \begin{cases} h(o) & , n = 0 \\ 2h(n) & ,n > 0 \end{cases} \qquad (B.29)$$

The function IFFT has been used. The input is:

HP: The filter's 1-D prototype.

18. TRANSFNC (Pg. 137)

Fww←TRANSFNC SIZE

This function produces a transformation matrix to design a FIR filter using McCLELLAN transformation. In this example the following function has been used:

$$F(\omega_1, \omega_2) = 1/2\,(-1 + \cos\omega_1 + \cos\omega_2 + \cos\omega_1 \cos\omega_2) \qquad (B.30)$$

In order to change $F(\omega_1, \omega_2)$ to a different transformation function, replace lines 10, 13, and 14 in the program with the new choices. (See example in Appendix C (Pg. 137). The input is:

SIZE: The number of samples of ω_1 or ω_2.

19. CHEB (Pg. 140)

Y←N CHEB X

This function calculates the Chebyshev Polynomials using the following recursion:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \qquad (B.31)$$

The inputs are:

a. N: The order of the desired polynomial

b. X: The argument.

WARNING: The combination of high order of Chebyshev polynomial and a large array to be evaluated will result in a long period of calculating time.

20. MCCLEL (Pg. 141)

```
Hww←a MCCLEL F
```

This function computes the 2-D frequency response using McCLELLAN transformation according to this equation:

$$H(\omega_1, \omega_2) = \sum_{n=0}^{N} a(n) T_n \left[F(\omega_1, \omega_2)\right] \qquad (B.32)$$

In order to indicate the calculating status, the function prompts the present calculating order of the polynomial. The inputs are:

a. a: The impulse response coefficients of the prototype filter

b. F: The transformation matrix (previously calculated using the function TRANSFNC).

APPENDIX C
FUNCTION LISTINGS AND EXAMPLES OF USE

```
      )LOAD 0 UTILITY
0 UTILITY SAVED 12/10/1987 16:37:45
               --------------------------------------------------------------
               | NAVAL POSTGRADUATE SCHOOL              MONTEREY.CALIFORNIA |
               |------------------------------------------------------------|
               |                     UTILITIES WORKSPACE                    |
               |                                                            |
               |                          THESIS BY                         |
               |                       Y.KATZIR.I.A.F.                      |
               |                ADVISOR:PROF. C.W. THERRIEN.                |
               | VERSION 1.0                                 SEPTEMBER 1987 |
               --------------------------------------------------------------
```

```
      )FNS
GETDATA INFORMATION     NORD    PUTDATA UNRD
```

```
      ∇NORD[⎕]∇
[0]   Z←N NORD P;S;I;T
[1]   ⍝
[2]   ⍝ A NORMAL DISTRIBUTION RANDOM VECTOR GENERATOR
[3]   ⍝
[4]    P←2↑P,1
[5]    Z←(N)⍴0
[6]    I←1
[7]   F10:T←2 UNRD 0.5 0.5
[8]    T←(2×T)-1
[9]    S←(T[1]*2)+T[2]*2
[10]   →F10×⍳S≥1
[11]   Z[I]←P[1]+P[2]×(T[1]×((¯2×⍟S)÷S)*0.5)
[12]   →F10×⍳(I←I+1)≤N
[13]   →0
[14]  ⍝
[15]  ⍝   SOURCE: GRAFSTAT
[16]  ⍝   N←NUMBER OF ELEMENTS IN THE VECTOR
[17]  ⍝   P←MEAN
[18]  ⍝   S←SIGMA (NOT SIGMA*2)
[19]  ⍝
```

```
      ∇UNRD[⎕]∇
[0]   R←N UNRD B
[1]   ⍝
[2]   ⍝  A UNIFORM DISTRIBUTION RANDOM VECTOR GENERATOR
[3]   ⍝
[4]    R←(B[1]-B[2])+((N?10000)×2×B[2])÷10000
[5]    →0
[6]   ⍝
[7]   ⍝  SOURCE: GRAFSTAT
[8]   ⍝  N←NUMBER OF ELEMENTS IN THE VECTOR
[9]   ⍝  B←THE LIMITS (2 NUMBERS)
[10]  ⍝
```

```
GAUSD←100 NORD 0 1


GAUSD
1.558274527 ¯2.444846636 ¯1.09846582 1.122362631 0.580943715
      ¯0.2714227577 0.4143999987 ¯0.9733729641 ¯1.021759908
      0.3179468935 1.515647616 0.7495965254 ¯0.5068186108 0.8851652886
      ¯0.2479112202 ¯0.7262416263 ¯0.4452313973 ¯0.6121927891
      ¯0.2086518602 0.5618385258 ¯1.064013245 0.3513946699 1.132830543
      0.1510880125 0.7028398161 ¯0.05231570544 2.018906718
      0.9240866379 ¯1.814329439 0.0351494428 ¯1.806312062 1.028100377
      0.3944470661 0.6391243468 0.8738735133 1.75599265 ¯0.3197284973
      ¯0.1366870369 0.6159939661 0.9774365509 ¯1.115427909
      ¯0.5496175242 0.03996605725 ¯2.484219992 1.158618634
      ¯1.026625814 1.153535117 ¯0.7858857486 0.6349969194 0.8198888762
      ¯0.1760331562 0.5627148978 ¯0.127309154 0.5538766995 ¯1.09726933
      ¯0.7311467753 1.404690913 ¯0.6203249336 0.2370824053
      ¯1.585608223 ¯0.4013458241 ¯0.7707101307 ¯0.2637646461
      0.9762420248 0.9775308888 1.169573915 0.1594895548 0.5001385714
      ¯1.05500756 ¯0.4509335123 1.271139261 0.8989835792 0.438888127
      ¯1.247429466 0.3241026886 0.3904619568 ¯0.4052863418
      0.2929672888 2.565225342 ¯0.4576884119 ¯1.611439324 ¯2.669649901
      ¯0.7594268721 ¯0.6750622611 ¯1.171739437 2.032501263
      0.9683033719 0.6699827805 0.4199990777 ¯2.871065022 1.686296301
      0.02722627758 ¯0.9020480504 ¯2.053207747 0.08940644014 2.0866291
      0.3651320115 0.8457720835 ¯0.1842246182 1.03036774


UNIFD←100 UNRD 0 100


UNIFD
3.22 ¯5.78 18.74 ¯10.04 43.86 91.44 ¯63.84 ¯19.84 ¯18.58 94.66 ¯51.14
      ¯46.78 ¯80.44 ¯23.98 ¯67.32 69.64 71.06 ¯82.22 ¯14.14 ¯37.6
      ¯66.84 70.92 ¯93.12 ¯30.38 18.1 92.06 50.94 ¯90.96 82.12 24.6
      29.6 83.32 ¯1.1 34.96 ¯92.3 61.72 ¯81.84 91.02 ¯77.52 3.34 ¯18.2
      26.18 ¯16.3 ¯93.02 ¯75.96 ¯14.68 66.88 44.58 ¯75.88 ¯2.82 ¯10.68
      ¯87.08 ¯34.64 81.86 ¯61.82 ¯68.26 48.36 35.22 6.46 ¯76.02 ¯84.16
      26.7 74.7 ¯62.86 ¯31.94 0.42 ¯42.96 96.16 33.74 ¯52.68 39.3
      ¯51.16 ¯8.2 ¯77.86 ¯60.26 ¯3.18 ¯13.76 16.38 64.04 ¯5.46 ¯24.86
      ¯31.52 ¯58.7 ¯35.88 ¯16.94 ¯47.26 73.98 ¯87.34 ¯94.42 ¯62.04
      ¯80.84 61.58 ¯89.98 57.78 ¯99.9 43.1 60.76 ¯32.04 ¯1.38 ¯89.56
```

```
    ∇PUTDATA[⎕]∇
[0]   F PUTDATA NAME;⎕IO;TIE;FA;NREC;C;FB;FC;BELL;I
[1]   ⍝
[2]   ⍝ FUNCTION TO WRITE DATA [F] (VECTOR,ARRAY,ETC.)IN FREE FORMAT
[3]   ⍝ INTO AN IBM P.C. D.O.S. FILE ['NAME']
[4]   ⍝
[5]    ⎕IO←I←1
[6]    ⎕NUNTIE ¯1
[7]    TIE←¯1
[8]   ⍝ CREATING A TIED FILE
[9]    NAME ⎕NCREATE TIE
[10]  ⍝ WRITING OUT THE SHAPE OF F
[11]   FA←'', 10 0 ⍕⍴F
[12]  ⍝ APPENDING THE SHAPE OF F TO THE FILE
[13]   (⎕TCLF,⎕TCNL,FA)⎕NAPPEND TIE
[14]  ⍝ RESHAPING F
[15]   NREC←⌈(×/⍴F)÷4
[16]   C←1+4|(×/(⍴F))-1
[17]   F←(NREC,4)⍴F
[18]  ⍝ WRITING OUT MOST OF THE ARRAY F
[19]   FB←⍕(¯1 0 ↓F)
[20]  LOOP:FB[I;(FB[I;]='¯')/⍳⍴FB[I;]]←'-'
[21]   →(NREC>I←I+1)/LOOP
[22]   (⎕TCLF,⎕TCNL,FB)⎕NAPPEND TIE
[23]  ⍝ WRITING OUT THE REMAINDER OF F AS THE LAST RECORD
[24]   FC←' ',⍕(C↑F[NREC;])
[25]   FC[(FC='¯')/⍳⍴FC]←'-'
[26]   (⎕TCLF,⎕TCNL,FC)⎕NAPPEND TIE
[27]  ⍝ UNTYING THE FILE
[28]   ⎕NUNTIE ¯1
[29]   BELL←⎕TCBEL
[30]   BELL
[31]   →0
[32]  ⍝
[33]  ⍝ THE FUNCTION USES NATIVES APL COMMANDS.
[34]  ⍝
[35]  ⍝ Y.KATZIR ,I.A.F., 6.1.87.
[36]  ⍝ ADVISOR: C.W. THERRIEN.
[37]  ⍝
```

```
GAUSD PUTDATA 'GAUS.DAT'

)OFF

C:\>TYPE GAUS.DAT


      100
  1.558274527   -2.444846636   -1.09846582      1.122362631
  0.580943715   -0.2714227577   0.4143999987   -0.9733729641
 -1.021759908    0.3179468935   1.515647616     0.7495965254
 -0.5068186108   0.3851652886  -0.2479112202   -0.7262416263
 -0.4452313973  -0.6121927891  -0.2086518602    0.5618385258
 -1.064013245    0.3513946699   1.132830543     0.1510880125
  0.7028398161  -0.05231570544  2.018906718     0.9240866379
 -1.814329439    0.0351494428  -1.806312062     1.028100377
  0.3944470661   0.6391243468   0.8738735133    1.75599265
 -0.3197284973  -0.1366870369   0.6159939661    0.9774365509
 -1.115427909   -0.5496175242   0.03996605725  -2.484219992
  1.158618634   -1.026625814    1.153535117    -0.7858857486
  0.6349969194   0.8198888762  -0.1760331562    0.5627148978
 -0.127309154    0.5538766995  -1.09726933     -0.7311467753
  1.404690913   -0.6203249336   0.2370824053   -1.585608223
 -0.4013458241  -0.7707101307  -0.2637646461    0.9762420248
  0.9775308888   1.169573915    0.1594895548    0.5001385714
 -1.05500756    -0.4509335123   1.271139261     0.8989835792
  0.438888127   -1.247429466    0.3241026886    0.3904619568
 -0.4052863418   0.2929672888   2.565225342    -0.4576884119
 -1.611439324   -2.669649901   -0.7594268721   -0.6750622611
 -1.171739437    2.032501263    0.9683033719    0.6699827805
  0.4199990777  -2.871065022    1.686296301     0.02722627758
 -0.9020480504  -2.053207747    0.08940644014   2.0866291
0.3651320115 0.8457720835 -0.1842246182 1.03036774
```

```
     ∇GETDATA[⎕]∇
  F←GETDATA NAME;⎕IO;DATA;M;SHAP;I;BELL
  ⍝
  ⍝ FUNCTION TO READ FREE FORMATED DATA [F] (VECTOR,ARRAY,ECT.)
  ⍝ FROM IBM P.C. D.O.S. FILE ['NAME']
  ⍝
   ⎕IO←I←0
   ⎕NUNTIE ¯1
  ⍝ TYING THE FILE
   NAME ⎕NTIE ¯1
  ⍝ FINDING THE SIZE OF THE FILE.
   M←⎕NSIZE ¯1
  ⍝ READING THE DATA INTO THE WORK SPACE
   F←⎕NREAD ¯1,82,M
   F[(F=⎕AV[45])/⍳⍴F]←⎕AV[253]
  ⍝ EXECUTING THE DATA AS AN APL EXPRESSION
   F←⍎(~(F=⎕AV[10])∨(F=⎕AV[13])∨(F=⎕AV[26]))/F
   ⎕NUNTIE ¯1
  LOOP:I←I+1
   →(I≥10)/ERROR
   DATA←I↓F
   SHAP←I↑F
   →((×/SHAP)≠(⍴DATA))/LOOP
   F←SHAP⍴DATA
   BELL←⎕TCBEL
   BELL
   →0
  ERROR:'FILE INCORECTLY FORMATED'
   →0
  ⍝
  ⍝ DATA HAVE BEEN CREATED BY APL,PL/I,FORTRAN,EDITOR ETC.
  ⍝
  ⍝ NUMBERS NEED NOT HAVE DECIMAL POINTS AND ARE DELIMITED BY SPACE.
  ⍝ FIRST RECORD IS SHAPE OF DATA.
  ⍝
  ⍝ Y.KATZIR,I.A.F., 6.1.87.
  ⍝ ADVISOR: C.W. THERRIEN.
  ⍝
```

```
      GAUSD←GETDATA 'GAUS.DAT'


      GAUSD
1.558274527 ¯2.444846636 ¯1.09846582 1.122362631 0.580943715 ¯0.27142275
      0.4143999987 ¯0.9733729641 ¯1.021759908 0.3179468935 1.515647616
      0.7495965254 ¯0.5068186108 0.8851652886 ¯0.2479112202 ¯0.726241626
      ¯0.4452313973 ¯0.6121927891 ¯0.2086518602 0.5618385258 ¯1.06401324
      0.3513946699 1.132830543 0.1510880125 0.7028398161 ¯0.05231570544
      2.018906718 0.9240866379 ¯1.814329439 0.0351494428 ¯1.806312062
      1.028100377 0.3944470661 0.6391243468 0.8738735133 1.75599265
      ¯0.3197284973 ¯0.1366870369 0.6159939661 0.9774365509 ¯1.115427909
      ¯0.5496175242 0.03996605725 ¯2.484219992 1.158618634 ¯1.026625814
      1.153535117 ¯0.7858857486 0.6349969194 0.8198888762 ¯0.1760331562
      0.5627148978 ¯0.127309154 0.5538766995 ¯1.09726933 ¯0.7311467753
      1.404690913 ¯0.6203249336 0.2370824053 ¯1.585608223 ¯0.4013458241
      ¯0.7707101307 ¯0.2637646461 0.9762420248 0.9775308888 1.169573915
      0.1594895548 0.5001385714 ¯1.05500756 ¯0.4509335123 1.271139261
      0.8989835792 0.438888127 ¯1.247429466 0.3241026886 0.3904619568
      ¯0.4052863418 0.2929672888 2.565225342 ¯0.4576884119 ¯1.611439324
      ¯2.669649901 ¯0.7594268721 ¯0.6750622611 ¯1.171739437 2.032501263
      0.9683033719 0.6699827805 0.4199990777 ¯2.871065022 1.686296301
      0.02722627758 ¯0.9020480504 ¯2.053207747 0.08940644014 2.0866291
      0.3651320115 0.8457720835 ¯0.1842246182 1.03036774
```

```
      )LOAD 0 EC3400
0 EC3400 SAVED 12/12/1987 21:23:25
            -----------------------------------------------------------
            | NAVAL POSTGRADUATE SCHOOL          MONTEREY,CALIFORNIA |
            |---------------------------------------------------------|
            |                   ECE -3400 WORKSPACE                   |
            |                                                         |
            |                       THESIS BY                         |
            |                    Y.KATZIR,I.A.F.                      |
            |             ADVISOR:PROF. C.W. THERRIEN.                |
            | VERSION 1.0                              NOVEMBER 1987 |
            -----------------------------------------------------------
```

```
      )FNS
BARTLETT          BPCOEFF BSCOEFF CC      COS     DIGFREQ FCC     FCOEFF  FFT
FLCV    FREQRES HAMMING HANNING HPCOEFF IDEALF  IFFT    INFORMATION     LCV
LPCOEFF MEAN    PI      SHIFT   SIN     UNSHIFT XCONJ   XEXPO   XMAGN   XPHAS
XSQRT   XTIME
```

```
      ∇DIGFREQ[⎕]∇
[0]   TC←FS DIGFREQ FC
[1]   ⍝
[2]   ⍝  CALCULATING THE DIGITAL FREQUENCY
[3]   ⍝
[4]    TC←(2×PI×FC)÷FS
[5]    →0
[6]   ⍝
[7]   ⍝  FS←SAMPLING FREQUENCY [Hz]
[8]   ⍝  FC←THE CORRESPONDING ANALOG FREQUNCY [Hz]
[9]   ⍝  TC:THE DIGITAL FREQUENCY [RAD.]
[10]  ⍝
[11]  ⍝  Y.KATZIR.I.A.F.,NOVEMBER 1987
[12]  ⍝
```

```
      FC←10000
      FS←50000

      TC←FS DIGFREQ FC

      TC
1.256637061

      TC÷PI
0.4
```

```
     ∇LPCOEFF[⎕]∇
[0]  hLP←n LPCOEFF TC;I;⎕IO
[1]  ⍝
[2]  ⍝  CALCULATING THE 'n' LOWPASS FIR FILTERS COEFFICIENTS USING FOURIER
[3]  ⍝  METHOD
[4]  ⍝
[5]   ⎕IO←1
[6]   n←n-1
[7]   n←ι(n+2)
[8]   hLP←nρ0
[9]   hLP←(1+(PI×(ιn)))×SIN((ιn)×TC)
[10]  hLP←(TC+PI),hLP
[11]  hLP←(n↑(⌽hLP)),hLP
[12]  ⎕IO←0
[13]  →0
[14] ⍝
[15] ⍝  THE COEFFICIENTS ARE FOR CAUSAL NONRECURSIVE FILTER.
[16] ⍝  THE COEFFICIENTS ARE NORMALIZED.
[17] ⍝  n←ODD NUMBER OF REQUIRED COEFFICIENTS (RECTANGULAR WINDOW)
[18] ⍝  TC←DIGITAL CUTOFF FREQUENCY [RAD.]
[19] ⍝
[20] ⍝  Y.KATZIR.I.A.F.,NOVEMBER 1987
[21] ⍝


      TC←0.4×PI
      n←21


      hLP←n LPCOEFF TC

      hLP
¯1.559907461E¯17 ¯0.0336367435 ¯0.02338723209 0.02672826525 0.05045511524
      ¯1.559907461E¯17 ¯0.07568267286 ¯0.06236595225 0.09354892838
      0.3027306915 0.4 0.3027306915 0.09354892838 ¯0.06236595225
      ¯0.07568267286 ¯1.559907461E¯17 0.05045511524 0.02672826525
      ¯0.02338723209 ¯0.0336367435 ¯1.559907461E¯17
```

```
    ∇HPCOEFF[⎕]∇
   hHP←n HPCOEFF TC;⎕IO;N;hLP
   ⍝
   ⍝  CALCULATING THE 'n' HIGHPASS FIR FILTER COEFFICIENTS USING FOURIER
   ⍝  METHOD
   ⍝
    ⎕IO←0
    N←n
    TC←PI-(TC)
    hLP←n LPCOEFF TC
    hHP←((-1)*⍳N)×hLP
    →0
   ⍝
   ⍝  THE COEFFICIENTS ARE FOR CAUSAL NONRECURSIVE FILTER.
   ⍝  THE MAGNITUDE IS NORMALIZED.
   ⍝  n←ODD NUMBER OF REQUIRED COEFFICIENTS (RECTANGULAR WINDOW)
   ⍝  TC←DIGITAL CUTOFF FREQUENCY [RAD.]
   ⍝
   ⍝  Y.KATZIR,I.A.F.,NOVEMBER 1987
   ⍝



      TC←0.4×PI
      n←21

      hHP←n HPCOEFF TC


      hHP
¯2.339861191E¯17 0.0336367435 0.02338723209 ¯0.02672826525 ¯0.05045511524
      ¯2.339861191E¯17 0.07568267286 0.06236595225 ¯0.09354892838
      ¯0.3027306915 0.6 ¯0.3027306915 ¯0.09354892838 0.06236595225
      0.07568267286 ¯2.339861191E¯17 ¯0.05045511524 ¯0.02672826525
      0.02338723209 0.0336367435 ¯2.339861191E¯17
```

```
      ∇BPCOEFF[⎕]∇
[0]   hBP←n BPCOEFF TUL;⎕IO;TC;TO;hLP
[1]   ⍝
[2]   ⍝  CALCULATING THE 'n' BANDPASS FIR FILTER COEFFICIENTS USING FOURIER
[3]   ⍝  METHOD
[4]   ⍝
[5]    ⎕IO←0
[6]    TC←(TUL[0]-TUL[1])+2
[7]    TO←(TUL[0]+TUL[1])+2
[8]    hLP←n LPCOEFF TC
[9]    n←n-1
[10]   n←⌊(n+2)
[11]   hLP←n↓hLP
[12]   hBP←(2×COS((⍳n+1)×TO))×hLP
[13]   hBP←(n↑(⌽hBP)),hBP
[14]   →0
[15] ⍝
[16] ⍝  THE COEFFICIENTS ARE FOR CAUSAL NONRECURSIVE FILTER.
[17] ⍝  THE COEFFICIENTS ARE NORMALIZED.
[18] ⍝  n←ODD NUMBER OF REQUIRED COEFFICIENTS (RECTANGULAR WINDOW)
[19] ⍝  TUL←DIGITAL UPPER AND LOWER CUTOFF FREQUENCIES (TWO ELEMENTS) [RAD]
[20] ⍝
[21] ⍝  Y.KATZIR,I.A.F.,NOVEMBER 1987
[22] ⍝


      TUL←2⍴0
      TUL[1]←0.6×PI
      TUL[2]←0.4×PI
      n←21

      hBP←n BPCOEFF TUL

      hBP
¯7.799537305E¯18 1.204383817E¯17 0.04677446419 ¯3.152585957E¯17 ¯0.1009102
      3.895627305E¯17 0.1513653457 ¯3.156840459E¯17 ¯0.1870978568
      1.207227699E¯17 0.2 1.207227699E¯17 ¯0.1870978568 ¯3.156840459E¯17
      0.1513653457 3.895627305E¯17 ¯0.1009102305 ¯3.152585957E¯17
      0.04677446419 1.204383817E¯17 ¯7.799537305E¯18
```

```
      ∇BSCOEFF[⎕]∇
[0]   hBS←n BSCOEFF TUL;⎕IO;hBP
[1]   ⍝
[2]   ⍝  CALCULATING THE 'n' BANDSTOP FIR FILTER COEFFICIENTS USING FOURIER
[3]   ⍝  METHOD
[4]   ⍝
[5]    ⎕IO←0
[6]    hBP←n BPCOEFF TUL
[7]    hBS←-hBP
[8]    n←n-1
[9]    hBS[n÷2]←1-hBP[n÷2]
[10]   →0
[11] ⍝  THE COEFFICIENTS ARE FOR CAUSAL NONRECURSIVE FILTER.
[12] ⍝  THE COEFFICIENTS ARE NORMALIZED.
[13] ⍝  n←ODD NUMBER OF REQUIRED COEFFICIENTS (RECTANGULAR WINDOW)
[14] ⍝  TUL←DIGITAL UPPER AND LOWER CUTOFF FREQUENCIES (TWO ELEMENTS) [RAD]
[15] ⍝
[16] ⍝  Y.KATZIR,I.A.F.,NOVEMBER 1987
[17] ⍝


      TUL[1]←0.4×PI
      TUL[2]←0.2×PI
      n←21

      hBS←n BSCOEFF TUL

      hBS
7.799537305E¯18 0.01284809274 ¯0.01445410434 ¯0.06997550689 ¯0.08163809137
      2.341241641E¯17 0.1224571371 0.1632761827 0.05781641735 ¯0.115632834
      0.8 ¯0.1156328347 0.05781641735 0.1632761827 0.1224571371
      2.341241641E¯17 ¯0.08163809137 ¯0.06997550689 ¯0.01445410434
      0.01284809274 7.799537305E¯18
```

```
     ∇HAMMING[⎕]∇
   W←HAMMING N;M;⎕IO
   ⍝
   ⍝  GENERATING THE 'N' WIDE HAMMING WINDOW
   ⍝
    ⎕IO←0
    W←Nρ0
    →((N+2)≠(⌈(N+2)))/ODD
  ⍝ EVEN NUMBER OF SAMPLES IN WINDOW
    M←(N+2)
    W←0.54+0.46×COS(((PI×2)×⍳M)+N)
    W←(⌽W),W
    →0
  ODD:M←(N+1)+2
    W←0.54+0.46×COS(((PI×2)×⍳M)+N)
    W←(⌽W),1↓W
    →0
  ⍝
  ⍝  THE WINDOW STARTS AT t=0 AND HAS 'N' SAMPLES.
  ⍝  N←NUMBER OF SAMPLES [VECTOR]
  ⍝
  ⍝  Y.KATZIR.I.A.P..NOVEMBER 1987
  ⍝


      N←31

      W←HAMMING N

      W
0.08236011124 0.1010959421 0.1378005566 0.1909712636 0.258431248 0.3374186
      0.4246998352 0.5167013823 0.6096567777 0.6997604163 0.7833234448
      0.8569247828 0.917551183 0.9627205933 0.990583773 1 0.990583773
      0.9627205933 0.917551183 0.8569247828 0.7833234448 0.6997604163
      0.6096567777 0.5167013823 0.4246998352 0.3374186903 0.258431248
      0.1909712636 0.1378005566 0.1010959421 0.08236011124
```

```
      ∇HANNING[⎕]∇
[0]  W←HANNING N;M;⎕IO
[1]  ⍝
[2]  ⍝ GENERATING THE 'N' WIDE HANNING WINDOW
[3]  ⍝
[4]   ⎕IO←0
[5]   W←Nρ0
[6]   →((N+2)≠(⌈(N+2)))/ODD
[7]  ⍝ EVEN NUMBER OF SAMPLES IN WINDOW
[8]   M←(N+2)
[9]   W←0.5+0.5×COS(((PI×2)×⍳M)+N)
[10]  W←(⌽W),W
[11]  →0
[12] ODD:M←(N+1)+2
[13]  W←0.5+0.5×COS(((PI×2)×⍳M)+N)
[14]  W←(⌽W),1↓W
[15]  →0
[16] ⍝
[17] ⍝  THE WINDOW STARTS AT t=0 AND HAS 'N' SAMPLES.
[18] ⍝  N←NUMBER OF SAMPLES [VECTOR]
[19] ⍝
[20] ⍝  Y.KATZIR,I.A.F.,NOVEMBER 1987
[21] ⍝


      N←31

      W←HANNING N

      W
2.565338304E¯3 0.0229303718 0.06282669193 0.1206209387 0.1939470087
      0.2798029242 0.3746737339 0.4746754156 0.5757138888 0.6736526264
      0.7644820052 0.8444834595 0.9103817206 0.9594789058 0.9897649706 1
      0.9897649706 0.9594789058 0.9103817206 0.8444834595 0.7644820052
      0.6736526264 0.5757138888 0.4746754156 0.3746737339 0.2798029242
      0.1939470087 0.1206209387 0.06282669193 0.0229303718 2.565338304E¯3
```

```
      ∇BARTLETT[⎕]∇
[0]  W←BARTLETT N;M;⎕IO
[1]  ⍝
[2]  ⍝  GENERATING THE 'N' WIDE BARTLETT (TRIANGULAR) WINDOW
[3]  ⍝
[4]   ⎕IO←0
[5]   ⍎←N, 0
[6]   →((N+2)≠(⌈(N+2)))/ODD
[7]  ⍝ EVEN NUMBER OF SAMPLES IN WINDOW
[8]   M←(N+2)
[9]   W←(⊖(1-(⍳M)+M)),1-(⍳M)+M
[10]  →0
[11] ODD:M←(N+1)+2
[12]  W←1-(⍳M)+M
[13]  W←(⊖W),1↓W
[14]  →0
[15] ⍝
[16] ⍝  THE WINDOW STARTS AT t=0 AND HAS 'N' SAMPLES.
[17] ⍝  N←NUMBER OF SAMPLES [VECTOR]
[18] ⍝
[19] ⍝  Y.KATZIR,I.A.F.,NOVEMBER 1987
[20] ⍝


      N←31

      W←BARTLETT N

      W
0.0625 0.125 0.1875 0.25 0.3125 0.375 0.4375 0.5 0.5625 0.625 0.6875 0.75
      0.8125 0.875 0.9375 1 0.9375 0.875 0.8125 0.75 0.6875 0.625 0.5625 0.5
      0.4375 0.375 0.3125 0.25 0.1875 0.125 0.0625
```

```
     ∇FREQRES[⎕]∇
[0]  HW←M FREQRES h;⎕IO;N;H
[1]  ⍝
[2]  ⍝  GENERATING FREQUENCY RESPONSE MAGNITUDE AND PHASE FROM TIME DOMAIN
[3]  ⍝  COEFFICIENTS
[4]  ⍝
[5]   ⎕IO←0
[6]   N←(M-⍴h)⍴0
[7]   h←h,N
[8]   H←FFT h
[9]   HW←(M+2)↑(XMAGN H)
[10]  PW←(M+2)↑(XPHAS H)
[11]  →0
[12] ⍝  THE FUNCTIONS FFT,XMAGN,XPHASE ARE USED.
[13] ⍝  IN ORDER TO RECEIVE A BETTER RESOLUTION THE COEFFICIENTS ARE ZERO-
[14] ⍝  PADDET TO 'M' SAMPLES.
[15] ⍝  M←NUMBER OF SAMPLES INCLUDING ZERO PADDING (USE ONLY RADIX TWO NO.
[16] ⍝  h←THE FILTER COEFFICIENTS [VECTOR]
[17] ⍝  HW:FREQUENCY RESPONSE MAGNITUDE (M+2 SAMPLES)
[18] ⍝  PW:FREQUENCY RESPONSE PHASE (M+2 SAMPLES) [RAD.]
[19] ⍝
[20] ⍝  Y.KATZIR,I.A.F,NOVEMBER 1987
[21] ⍝


      hLP←21 LPCOEFF (0.4×PI)

      HLP←128 FREQRES hLP

      HLP
0.9567807992 0.9619585014 0.9763182673 0.9965906571 1.018111173 1.03582952
      1.045421125 1.044271236 1.032124069 1.01125183 0.986091438 0.9624009
      0.9460845406 0.9419036347 0.9523233652 0.9767236974 1.011145443
      1.048645824 1.080227975 1.096202394 1.087755987 1.048461283 0.975462
      0.8701304604 0.7380554233 0.5883839646 0.4325882327 0.282876649
      0.1505027492 0.04424706035 0.03068671843 0.0731944037 0.08614789716
      0.07567134118 0.04996228114 0.01787457278 0.01248921845 0.0349417940
      0.04595168274 0.04486445606 0.03360579506 0.01595981635 3.413311717E¯3
      0.02001949514 0.03043376941 0.03290460379 0.0275668884 0.01625216492
      1.963679943E¯3 0.01185437487 0.02209076454 0.02662206686 0.024719091
      0.01713279563 5.857126075E¯3 6.366569896E¯3 0.01668743018 0.02278343
      0.02336291733 0.01842504363 9.228990525E¯3 2.014864013E¯3 0.01265918
      0.02022375085
```

```
     ∇IDEALF[⎕]∇
 H←TYPE IDEALF TC;⎕IO
 ⍝
 ⍝  GENERATING IDEAL FREQUENCY RESPONSE
 ⍝
  ⎕IO←0
  H←TC[0]⍴0
  →((×/(TYPE='LP'))∨(×/(TYPE='HP')))/LP
  →((×/(TYPE='BP'))∨(×/(TYPE='BS')))/BP
  'IMPROPER ARGUMENT.USE ONLY: LP. HP . BP .OR  BS .
  →0
LP:H[⍳(⌊((TC[0]+2)×TC[1])+1)]←1
  H←(TC[0]+2)↑H
  →(×/(TYPE='LP'))/CONTINUE
  H←~H
  →CONTINUE
BP:H[(⍳(⌊((TC[0]+2)×(TC[1]-TC[2]))))+(⌊(((TC[0]+2)×TC[2])+1))]←1
  H←(TC[0]+2)↑H
  →(×/(TYPE='BP'))/CONTINUE
  H←~H
CONTINUE:H←H,⌽H
  →0
⍝
⍝  'TYPE'←THE TYPE OF FILTER TO BE GENERATED [LP.HP.BP.BS]
⍝  TC[0]←NUMBER OF SAMPLES (USE ONLY RADIX TWO NO.)
⍝  TC[1]←THE CUTOFF DIGITAL FREQUENCY FOR LP OR HP FILTER OR
⍝  THE UPPER DIGITAL FREQUENCY FOR BP OR BS FILTERS
⍝  TC[2]←THE LOWER DIGITAL FREQUENCY FOR BP OR BS FILTERS
⍝  THE FREQUENCIES SHOULD BE A FRACTION OF PI [RAD.].
⍝
⍝  Y.KATZIR ,I.A.F.,NOVEMBER 1987
⍝
```

```
TC[1]←128
TC[2]←0.6
H←'HP' IDEALF TC

      H
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

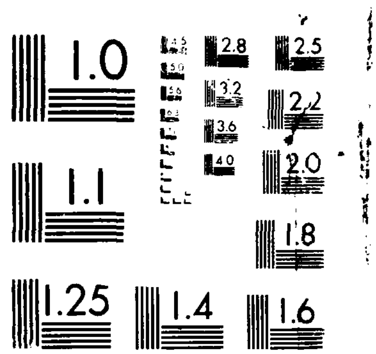
```
TC←3ρ0
TC[1]←128
TC[2]←0.3
TC[3]←0.12

H←'BP' IDEALF TC

      H
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
      1 1 1 1 1 1 0 0 0 0 0 0 0 0
```

```
      ∇FCOEFF[⎕]∇
[0]   h←n FCOEFF HW;⎕IO;H;m;SIZE
[1]   ⍝
[2]   ⍝  CALCULATING THE 'n' FILTER COEFFICIENTS  USING IFFT
[3]   ⍝
[4]    ⎕IO←0
[5]    n←n-1
[6]    SIZE←⌊(n÷2)
[7]    m←SIZE+1
[8]    H←IFFT HW
[9]    h←(1,m)↑H
[10]   h←m⍴h
[11]   h←(SIZE↑(⌽h)),h
[12]   →0
[13] ⍝
[14] ⍝  THE COEFFICIENTS ARE FOR CAUSAL NONRECURSIVE FILTER.
[15] ⍝  n←ODD NUMBER OF REQUIRED COEFFICIENTS (RECTANGULAR WINDOW)
[16] ⍝  HW← THE IDEAL FILTER SAMPLES (RADIX TWO)
[17] ⍝
[18] ⍝  Y.KATZIR.I.A.F.,NOVEMBER 1987
[19] ⍝


      H
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
      1 1 1 1 1 1 0 0 0 0 0 0 0 0

      n←21

      h←n FCOEFF H


      h
0.02511120991 0.04315473177 0.03628637547 ¯6.290075645E¯3 ¯0.07065358471
      ¯0.1217862264 ¯0.1233903877 ¯0.06216894475 0.03973030337 0.133830098
      0.171875 0.1338300989 0.03973030337 ¯0.06216894475 ¯0.1233903877
      ¯0.1217862264 ¯0.07065358471 ¯6.290075645E¯3 0.03628637547 0.0431547
      0.02511120991
```

```
      )LOAD 0 EC3410
0 EC3410 SAVED 12/11/1987 12:48:01
              ------------------------------------------------------------
              | NAVAL POSTGRADUATE SCHOOL            MONTEREY,CALIFORNIA |
              |----------------------------------------------------------|
              |                   ECE -3410 WORKSPACE                    |
              |                                                          |
              |                        THESIS BY                         |
              |                     Y.KATZIR.I.A.F.                      |
              |            ADVISOR:PROF. C.W. THERRIEN.                  |
              | VERSION 1.0                               SEPTEMBER 1987 |
              ------------------------------------------------------------

      )FNS
CC      COS     COV     FCC     FFT     FLCV    IFFT    INFORMATION     LCV
LPFP    MEAN    PI      PSE     PSE∆    SACF    SIN     XCONJ   XEXPO   XMAGN
XPHAS   XSQRT   XTIME
```

AD-A195 193 PC SOFTWARE FOR THE TEACHING OF DIGITAL SIGNAL PROCESSING(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA Y KATZIR MAR 88 2/2

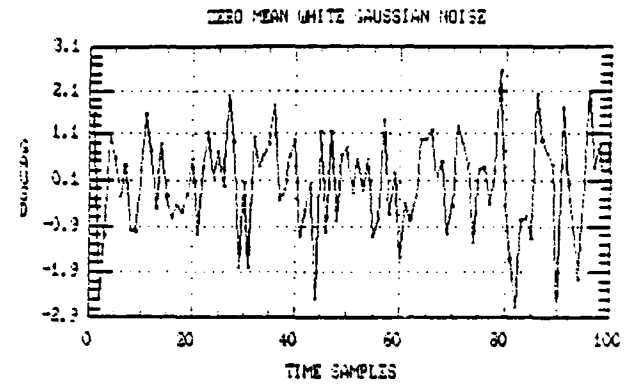
UNCLASSIFIED F/G 12/5 NL

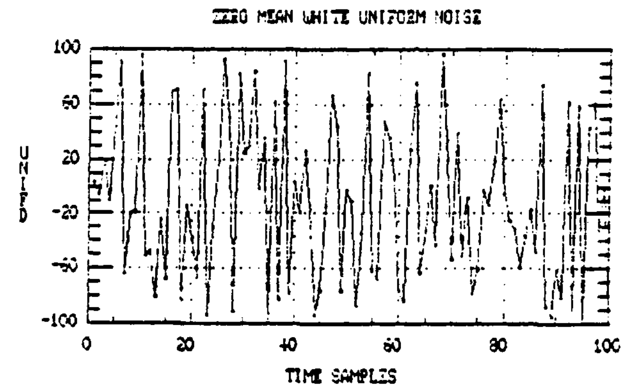


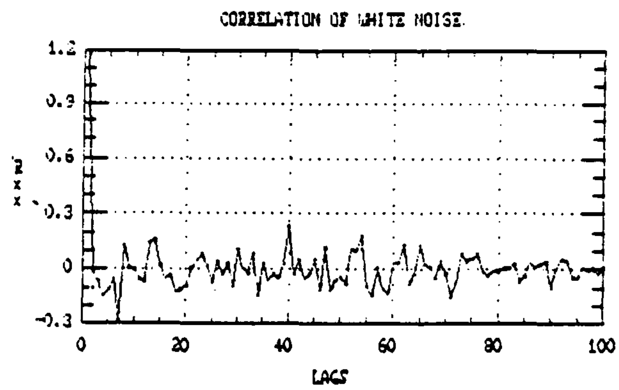
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963

```
      ∇SACF[⎕]∇
    [0]  R←LAG SACF DAT;⎕IO;MEAN;N
    [1]  ⍝
    [2]  ⍝ SAMPLE AUTOCORRELATION FUNCTION (UNBIASED)
    [3]  ⍝
    [4]   ⎕IO←K←0
    [5]   N←⍴DAT
    [6]   R←LAG⍴0
    [7]   DAT←DAT,R
    [8]  LOOP:R[K]←(+/DAT×(-K)⌽DAT)÷N
    [9]   →(LAG>K←K+1)/LOOP
    [10]  →0
    [11] ⍝
    [12] ⍝ Y.KATZIR,I.A.F.,JULY 1987
    [13] ⍝
    [14] ⍝ DAT←DATA VECTOR (ONE DIMENSIONAL)
    [15] ⍝ LAG←NO. OF LAGS (MAX. 300)
    [16] ⍝
```

```
      Rxx←100 SACF GAUSD


      Rxx
1.1816 ¯0.011604 ¯0.066841 ¯0.14886 ¯0.11086 ¯0.050135 ¯0.28244
      0.12983 0.011973 ¯2.0869E¯3 ¯0.052813 ¯0.066688 0.1414 0.16079
      0.021921 ¯0.050052 ¯0.03314 ¯0.12319 ¯0.11111 ¯0.087361
      7.9253E¯3 0.047914 0.082207 0.022012 ¯0.072971 0.047907
      ¯0.028347 0.03337 ¯0.091018 0.11214 4.3453E¯3 ¯0.020604 0.08519
      ¯0.13963 0.038409 ¯0.059126 ¯0.023359 ¯0.04759 0.03288 0.23321
      ¯0.021454 0.051991 ¯0.049765 ¯0.025113 0.059572 ¯0.11704 0.11551
      ¯0.11528 ¯0.058985 ¯0.038291 ¯0.077532 0.11274 0.10221 0.18426
      ¯0.09541 ¯0.14171 8.3982E¯3 ¯0.10611 ¯0.13464 0.037944 0.035897
      0.13416 ¯0.081343 ¯0.028463 0.13128 0.026911 0.011188 ¯0.048038
      0.048817 ¯0.023835 ¯0.14964 ¯0.071093 0.084411 0.04223 0.055516
      0.08221 ¯0.011808 ¯0.03549 ¯4.7687E¯3 ¯0.011637 0.010587
      7.3499E¯3 0.033257 ¯0.065615 ¯0.036728 0.038447 0.014501
      0.029666 0.042598 ¯0.096824 5.4581E¯3 0.052421 0.047676
      ¯0.044033 ¯0.049402 0.016731 ¯1.8189E¯3 6.5336E¯3 ¯0.028012
      0.015022
```


ZERO MEAN WHITE GAUSSIAN NOISE
TIME SAMPLES


ZERO MEAN WHITE UNIFORM NOISE
TIME SAMPLES


CORRELATION OF WHITE NOISE
LAGS

```
      ∇MEAN[⎕]∇
[0]   M←MEAN F
[1]   ⍝ CALCULATE THE MEAN
[2]    M←(+/+/F)÷×/⍴F
[3]   ⍝
[4]   ⍝ SOURCE:MDSP W.SPACE
[5]   ⍝ F←DATA VECTOR
[6]   ⍝ M:THE MEAN (NUMBER)
[7]   ⍝
```

```
      X←⍳100

      X
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
      28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
      50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71
      72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93
      94 95 96 97 98 99 100

      XX←MEAN X

      XX
50.5
      X←X-XX
      X
¯49.5 ¯48.5 ¯47.5 ¯46.5 ¯45.5 ¯44.5 ¯43.5 ¯42.5 ¯41.5 ¯40.5 ¯39.5
      ¯38.5 ¯37.5 ¯36.5 ¯35.5 ¯34.5 ¯33.5 ¯32.5 ¯31.5 ¯30.5 ¯29.5
      ¯28.5 ¯27.5 ¯26.5 ¯25.5 ¯24.5 ¯23.5 ¯22.5 ¯21.5 ¯20.5 ¯19.5
      ¯18.5 ¯17.5 ¯16.5 ¯15.5 ¯14.5 ¯13.5 ¯12.5 ¯11.5 ¯10.5 ¯9.5 ¯8.5
      ¯7.5 ¯6.5 ¯5.5 ¯4.5 ¯3.5 ¯2.5 ¯1.5 ¯0.5 0.5 1.5 2.5 3.5 4.5 5.5
      6.5 7.5 8.5 9.5 10.5 11.5 12.5 13.5 14.5 15.5 16.5 17.5 18.5
      19.5 20.5 21.5 22.5 23.5 24.5 25.5 26.5 27.5 28.5 29.5 30.5 31.5
      32.5 33.5 34.5 35.5 36.5 37.5 38.5 39.5 40.5 41.5 42.5 43.5 44.5
      45.5 46.5 47.5 48.5 49.5

      XX←MEAN X

      XX
0
```

```
      ∇COV[⎕]∇
[0]   K←COV R;⎕IO;N;I
[1]   ⍝
[2]   ⍝  GENERATE TOEPLITZ COVARIANCE MATRIX FROM COVARIANCE FUNCTION
[3]   ⍝
[4]    ⎕IO←I←0
[5]    N←⍴R
[6]    K←(N,N)⍴0
[7]    R←(⌽1↓R),R
[8]   LOOP:K[I;]←(-N)↑(-I)↓R
[9]    →(N>I←I+1)/LOOP
[10]  ⍝
[11]  ⍝  SOURCE: MDSP W.SPACE
[12]  ⍝  R←THE CORRELATION FUNCTION (VECTOR)
[13]  ⍝
```

```
      RCL1←3 SACF CL1


      RCL1
7.3328 6.8337 6.4035


      CMCL1←COV RCL1


      GMCL1
 7.3328 6.8337 6.4035
 6.8337 7.3328 6.8337
 6.4035 6.8337 7.3328
```

```
      ∇LPFP[⎕]∇
[0]   a←P LPFP DATA;⎕IO;R;K;MDATA;M
[1]   ⍝
[2]   ⍝ GENERATE THE Pth ORDER LINEAR PREDICTION FILTER PARAMETERS
[3]   ⍝ AND THE PREDICTION ERROR VARIANCE
[4]   ⍝
[5]    ⎕IO←0
[6]    a←(P+1)ρ0
[7]    M←MEAN DATA
[8]    MDATA←DATA-M
[9]    R←(P+1)SACF MDATA
[10]   K←COV R
[11]   a←1,a←-(1↓R)⌹(P,P)↑K
[12]   PERR←+/(a×R)
[13]   →0
[14]  ⍝
[15]  ⍝ THE FUNCTION SOLVES NORMAL EQUATIONS USING THE SACF,COV,AND MEAN
[16]  ⍝ FUNCTIONS.
[17]  ⍝
[18]  ⍝ Y.KATZIR,I.A.F.,AUGUST 87.
[19]  ⍝
[20]  ⍝ a: FILTER COEFFICIENTS
[21]  ⍝ PERR: THE ESTIMATE PREDICTION ERROR VARIANCE
[22]  ⍝ P←THE FILTER ORDER
[23]  ⍝ DATA←THE DATA VECTOR
[24]  ⍝



      a←3 LPFP CL1


      a
1 ¯0.897346235 ¯0.017308669 ¯0.02113346634

      PERR
0.9625826774
```

```
      ∇CC[⎕]∇
[0]   Y←X CC H;⎕IO;N;I;NN;SNN
[1]   ⍝
[2]   ⍝ 1-D CIRCULAR CONVOLUTION
[3]   ⍝
[4]    ⎕IO←I←0
[5]    SNN←⍴(NN←⍴H)
[6]    N←⍴X
[7]    →(N≠NN)/ERROR
[8]    →(SNN≠1)/ERROR
[9]    Y←N⍴0
[10]  LOOP:Y[I]←+/X×(-1+I)⌽⌽H
[11]   →(N>I←I+1)/LOOP
[12]   →0
[13]  ERROR:'ARGUMENTS ARE NOT THE SAME SIZE OR 1-D.
[14]   →0
[15]  ⍝
[16]  ⍝ Y.KATZIR.I.A.F,AUGUST 1987
[17]  ⍝
[18]  ⍝ ARGUMENTS NEED TO BE THE SAME SIZE AND 1-D.
[19]  ⍝ X AND H←ARGUMENTS TO BE CONVOLVED (VECTOR)
[20]  ⍝
```

```
      X
3 2 1

      H
3 2 1

      Y←X CC H

      Y
13 13 10

      LONG
1 2 3 4 5 6 7 8 9 10

      X CC LONG
ARGUMENTS ARE NOT THE SAME SIZE OR 1-D.
```

```
      ∇FCC[⎕]∇
[0]   Y←X FCC H;⎕IO;N;NN;SNN;A
[1]   ⍝
[2]   ⍝ FAST 1-D CIRCULAR CONVOLUTION
[3]   ⍝
[4]    ⎕IO←0
[5]    SNN←⍴(NN←⍴H)
[6]    N←⍴X
[7]    →(NN>80)/STATM
[8]    →(N≠NN)/ERROR
[9]    →(SNN≠1)/ERROR
[10]   A←(N,N)⍴⌽H
[11]   A←(-⍳N)⌽A
[12]   Y←1⌽(A+.×X)
[13]   →0
[14]  ERROR:'ARGUMENT ARE NOT THE SAME SIZE OR 1-D.'
[15]   →0
[16]  STATM:'PLEASE USE CC OR LCV FUNCTIONS.'
[17]   →0
[18]  ⍝
[19]  ⍝ Y.KATZIR.I.A.F.,AUGUST 1987
[20]  ⍝
[21]  ⍝ ARGUMENTS NEED TO BE THE SAME SIZE AND 1-D.
[22]  ⍝ LENGTH OF ARGUMENT VECTOR NEED TO BE SHORT.
[23]  ⍝ X AND H←ARGUMENT TO BE CONVOLVED
[24]  ⍝



      X
3 2 1

      H
3 2 1

      Z←X FCC H

      Z
13 13 10
```

```
      ∇LCV[⎕]∇
[0]   Y←X1 LCV H1;⎕IO;N1;N2;N;XZ;HZ;X;H
[1]   ⍝
[2]   ⍝ 1-D LINEAR CONVOLUTION
[3]   ⍝
[4]    ⎕IO←0
[5]    N1←⍴X1
[6]    N2←⍴H1
[7]    N←N1+N2-1
[8]    XZ←N↑X1
[9]    HZ←N↑H1
[10]   Y←XZ CC HZ
[11]   →0
[12]  ⍝
[13]  ⍝ LINEAR CONVOLUTION USING CC FUNCTION
[14]  ⍝
[15]  ⍝ Y.KATZIR,I.A.F.,AUGUST 1987
[16]  ⍝
[17]  ⍝ X AND H←THE ARGUMENTS TO BE CONVOLVED (VECTORS)
[18]  ⍝


      X
3 2 1
      H
3 2 1
      Y←X LCV H
      Y
9 12 10 4 1


      LONG
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
      30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
      54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77
      78 79 80 81 82 83 84 85 86 87 88 89 90
      Y←X LCV LONG
      Y
3 8 14 20 26 32 38 44 50 56 62 68 74 80 86 92 98 104 110 116 122 128 134 140
      146 152 158 164 170 176 182 188 194 200 206 212 218 224 230 236 242 248
      254 260 266 272 278 284 290 296 302 308 314 320 326 332 338 344 350 356
      362 368 374 380 386 392 398 404 410 416 422 428 434 440 446 452 458 464
      470 476 482 488 494 500 506 512 518 524 530 536 269 90
```

```
      ∇FLCV[⎕]∇
[0]   Y←X1 FLCV H1;⎕IO;N1;N2;N;XZ;HZ;X;H
[1]   ⍝
[2]   ⍝ FAST 1-D LINEAR CONVOLUTION
[3]   ⍝
[4]    ⎕IO←0
[5]    N1←⍴X1
[6]    N2←⍴H1
[7]    N←N1+N2-1
[8]    XZ←N↑X1
[9]    HZ←N↑H1
[10]   Y←XZ FCC HZ
[11]   →0
[12]  ⍝
[13]  ⍝ FAST LINEAR CONVOLUTION USING FCC FUNCTION
[14]  ⍝
[15]  ⍝ Y.KATZIR.I.A.F.,AUGUST 1987
[16]  ⍝
[17]  ⍝ X AND H←THE ARGUMENTS TO BE CONVOLVED (VECTORS)
[18]  ⍝


      X
3 2 1

      H
3 2 1

      Y←X FLCV H

      Y
9 12 10 4 1



      Y←X FLCV LONG
PLEASE USE CC OR LCV FUNCTIONS.
VALUE ERROR
FLCV[10] Y←XZ FCC HZ
              ^
```

```
      ∇FFT[⎕]∇
[0]   z←FFT x;k;n;r;⎕IO
[1]    ⎕IO←0
[2]    →(r←(2∊ρρx)∧2∊1↑ρx)↑8
[3]    →(1∊ρk)↑7-r←25>k←⌊0.5×ρx
[4]    'NOT A COMPLEX VECTOR'
[5]    →0
[6]    →8,ρx←x,[¯0.5]0
[7]    x←⍉(k,2)ρx÷2
[8]    →(20≥k← ¯1 2 4 8 16 32 64 128 256 512 1024 2048 4096 8192 16384 3276
[9]    →(1↓ρz←x)↓ 0 0
[10]   'NOT A POWER-OF-2 ORDER'
[11]   →-⎕EX 'z'
[12]   z←(1 2 ,kρ2)ρx
[13]   →k↓ 0 28 23
[14]   →(2∊⎕NC 'wt')↓16
[15]   →(k∉ρρx←wt)↑20
[16]   n←0,-⌽1↓x←2○(○2÷n)×⍳n÷4
[17]   wt←x←(kρ2)ρ(x,n),n,-x
[18]   →20
[19]   x←⍎(kρ'x[;;;;;;;;;;;;;;;;;;'),'0]'
[20]   n←-/[2]z
[21]   k←ρz←+/[2]z
[22]   →(4<k←ρρz←z,[0](-/[1]n×kρx),[0.5]+/[1]n×kρ⊖x)↑19
[23]   n←-/[2]z
[24]   z←+/[2]z
[25]   n[;0;1]←-n[;0;1]
[26]   n[; 1 0 ;1]←n[;;1]
[27]   z←z,[0]n
[28]   z←⍉(+/z),[n←0]-/z
[29]   →⍳r
[30]   n←¯1⌽⌽z
[31]   n[1;]←-n[1;]
[32]   x←n-z
[33]   n←¯1↑ρz←z+n
[34]   k←0,⌽1↓r←2○(○÷n)×⍳n÷2
[35]   r←(ρz)ρk,r,-r
[36]   x←(+⌿x×r),[¯0.5]-⌿r×⊖x
[37]   z←(z+x),z-x
[38]   →0
[39] ⍝
[40] ⍝ APL-STYLE FAST FOURIER TRANSFORM
[41] ⍝  x←CAN BE EITHER REAL (A VECTOR) OR COMPLEX (2-ROW MATRIX)
[42] ⍝  OPTIMIZED FOR SPEED
[43] ⍝  Z:THE OUTPUT IS COMPLEX (REAL AND IMAGE PARTS).
[44] ⍝  SOURCE:PAUL PENFIELD,JR.6/13/79
[45] ⍝
```

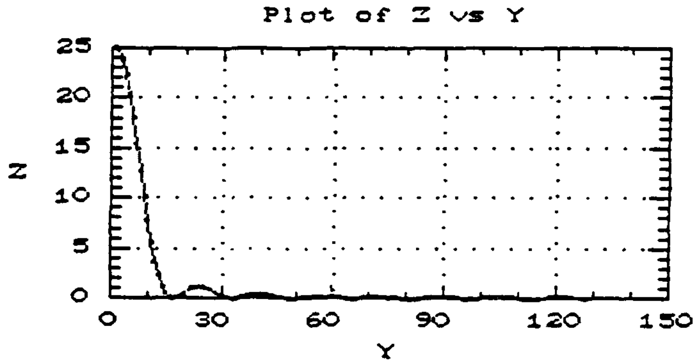
```
      ∇IFFT[⎕]∇
[0]   z←IFFT x;k;n;r;⎕IO
[1]    ⎕IO←0
[2]    →(r←(2∊⍴⍴x)∧2∊1↑⍴x)↑8
[3]    →(1∊⍴k)↑7-r←25>k←⌊0.5×⍴x
[4]    'NOT A COMPLEX VECTOR'
[5]    →0
[6]    →8,⍴x←x,[¯0.5]0
[7]    x←⍉(k,2)⍴x÷4
[8]    →(20≥k← ¯1 2 4 8 16 32 64 128 256 512 1024 2048 4096 8192 16384 3276
[9]    →(1↓⍴z←x)↓ 0 0
[10]   'NOT A POWER-OF-2 ORDER'
[11]   →-⎕EX 'z'
[12]   z←(1 2 ,k⍴2)⍴x÷n
[13]   →k↓ 0 28 23
[14]   →(2∊⎕NC 'wt')↓16
[15]   →(k∊⍴⍴x←wt)↑20
[16]   n←0,-⌽1↓x←2○(○2÷n)×⍳n÷4
[17]   wt←x←(k⍴2)⍴(x,n),n,-x
[18]   →20
[19]   x←⍎(k⍴'x[;;;;;;;;;;;;;;;;;;;;').'0]'
[20]   n←-/[2]z
[21]   k←⍴z←+/[2]z
[22]   →(4<k←⍴⍴z←z,[0](+/[1]n×k⍴x),[0.5]-/[1]n×k⍴⊖-x)↑19
[23]   n←-/[2]z
[24]   z←+/[2]z
[25]   n[;1;1]←-n[;1;1]
[26]   n[; 1 0 ;1]←n[;;1]
[27]   z←z,[0]n
[28]   z←⍉(+/z),[n←0]-/z
[29]   →⍳r
[30]   n←¯1⌽⌽z
[31]   n[1;]←-n[1;]
[32]   x←z-n
[33]   n←¯1↑⍴z←z+n
[34]   k←0,⌽1↓r←2○(○÷n)×⍳n÷2
[35]   r←(⍴z)⍴k,r,-r
[36]   x←(-≠x×r),[¯0.5]+≠x×⊖r
[37]   z←(z+x),z-x
[38]   →0
[39] ⍝
[40] ⍝  APL-STYLE INVERSE FOURIER TRANSFORM
[41] ⍝  x←CAN BE EITHER REAL (A VECTOR) OR COMPLEX (2-ROW MATRIX)
[42] ⍝  OPTIMIZED FOR SPEED
[43] ⍝  SOURCE:PAUL PENFIELD,JR. 6/13/79
[44] ⍝
```

```
      X
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

      Y←FFT X

      Y
 16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

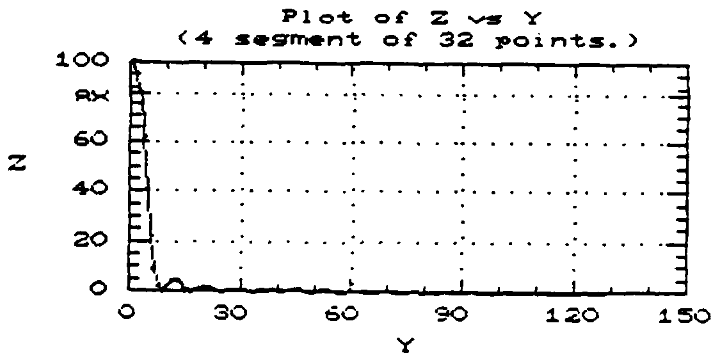
```
      Y
 16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

      XX←IFFT Y

      XX
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

      XX←XX[0;]

      XX
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
      ∇PSE[⎕]∇
[0]   INW←ZP PSE DATA;⎕IO:N;XN;XW
[1]   ⍝
[2]   ⍝  COMPUTING THE POWER SPECTRUM ESTIMATION (PERIODOGRAM)
[3]   ⍝
[4]    ⎕IO←0
[5]    N←(⍴DATA)+ZP
[6]    N←2*⌈(2⍟N)
[7]    XN←DATA,((N-(⍴DATA))⍴0)
[8]    XW←FFT XN
[9]    INW←(+⌿(XW*2))÷N
[10]   →0
[11] ⍝
[12] ⍝  COMPUTING THE POWER SPECTRUM ESTIMATION (PERIODOGRAM) USING FFT
[13] ⍝  FUNCTION
[14] ⍝
[15] ⍝  Y.KATZIR ,I.A.F.,AUGUST 1987
[16] ⍝
[17] ⍝  DATA←THE DATA VECTOR (1-D)
[18] ⍝  ZP←LENGTH OF ZERO PADDING FOR SMOOTH PLOT
[19] ⍝  INW:THE OUTPUT-SX(ω)
[20] ⍝


      ∇PSEB[⎕]∇
[0]   INBW←SEG PSEB DATA;⎕IO;M;MM;XW;XN;L;K
[1]   ⍝
[2]   ⍝  COMPUTING THE POWER SPECTRUM ESTIMATION USING BARTLETT WINDOW
[3]   ⍝
[4]    ⎕IO←0
[5]    M←⌊((⍴DATA)÷SEG)
[6]    XN←(M,SEG)⍴0
[7]    ZP←256-SEG
[8]    MM←(1↓⍴XN)+ZP
[9]    XW←(M,MM)⍴0
[10]   L←K←0
[11] LOOP1:XN[K;⍳SEG]←DATA[(SEG×K)+⍳SEG]
[12]   →(M>K←K+1)/LOOP1
[13] LOOP2:XW[L;]←ZP PSE XN[L;]
[14]   →(M>L←L+1)/LOOP2
[15]   INBW←(+⌿XW)÷M
[16]   →0
[17] ⍝
[18] ⍝  Y.KATZIR,I.A.F.,AUGUST 1987
[19] ⍝
[20] ⍝  SEG←LENGTH OF SEGMENTS
[21] ⍝  DATA←THE DATA VECTOR (I-D)
[22] ⍝  USING THE PSE FUNCTION (+FFT).
[23] ⍝  THE FUNCTION PADS THE SEGMENT WITH ZEROS UP TO 256 POINTS.
[24] ⍝
```

```
      X
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5

      Y←240 PSE X
      Z←128↑Z
      Y←ι128
```




```
      X
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
      5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
      5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
      5 5 5 5 5 5 5 5 5 5 5 5 5 5

      Z←32 PSEB X
      ⍴Z
256
      Z←128↑Z
```



```
      ∇XSQRT[⎕]∇
[0]   XZ←XSQRT XR
[1]   ⍝
[2]   ⍝ COMPLEX SQUARE ROOT (XR*0.5)
[3]   ⍝
[4]    XZ←(0.5-0∧.>XR)×(2×((+/XR×XR)*0.5)+( 1 0 +.×XR)*0.5
[5]    XZ←(0∧.> 1 0 /XR)⊖XZ,[⎕IO-0.5](0 1 +.×XR)÷2×XZ+XZ=0
```

```
      X
0.5 0.866

      XSQRT X
0.86602 0.49999
```

```
      ∇XTIME[⎕]∇
[0]   XZ←XL XTIME XR
[1]   ⍝
[2]   ⍝ COMPLEX TIMES (XL×XR)
[3]   ⍝
[4]    XZ←((1 0 /XL)× 1 0 /XR)-(0 1 /XL)× 0 1 /XR
[5]    XZ←XZ,[⎕IO]((1 0 /XL)× 0 1 /XR)+(0 1 /XL)× 1 0 /XR
```

```
      X
0.5 0.866

      X XTIME X
¯0.49996 0.866
```

```
      ∇XMAGN[⎕]∇
[0]   Z←XMAGN XR
[1]   ⍝
[2]   ⍝ COMPLEX MAGNITUDE (|XR) --RETURN IS REAL
[3]   ⍝
[4]    Z←(+/XR*2)*0.5
```

```
      X
0.5 0.866

      XMAGN X
0.99998
```

```
      ∇XPHAS[⎕]∇
[0]   Z←XPHAS XR
[1]   ⍝
[2]   ⍝ COMPLEX PHASE ANGLE --RETURN IS REAL
[3]   ⍝
[4]    Z← 0 1 +.×XR
[5]    Z←(×Z+Z=0)×(0∨.≠XR)×¯2○(1 0 +.×XR)+(0∧.=XR)+(+/XR×XR)*0.5
```

```
      X
0.5 0.866

      XPHAS X
1.0472
```

```
      ∇XCONJ[⎕]∇
[0]   XZ←XCONJ XR
[1]   ⍝
[2]   ⍝ COMPLEX CONJUGATE (+XR)
[3]   ⍝
[4]    XZ←(1 0 ≠XR),[⎕IO]- 0 1 ≠XR
```

```
      X
0.5 0.866

      XCONJ X
0.5 ¯0.866
```

```
      ∇XEXPO[⎕]∇
[0]   XZ←XEXPO XR
[1]   ⍝
[2]   ⍝ COMPLEX EXPONENTIAL (*XR)
[3]   ⍝
[4]    XZ←((⍴XR)⍴* 1 0 ≠XR)× 2 1 ∘.○ 0 1 +.×XR
```

```
      X
0.5 0.866

      XEXPO X
1.0682 1.2559
```

```
      ∇PI[⎕]∇
 [0]  Y←PI
 [1]   Y←○1

      PI
3.1416

      ∇COS[⎕]∇
 [0]  Y←COS X
 [1]   Y←2○X

      COS (PI÷4)
0.70711

      ∇SIN[⎕]∇
 [0]  Y←SIN X
 [1]   Y←1○X

      SIN (PI÷2)
1
```

```
      )LOAD 0 EC4440
0 EC4440 SAVED 2/23/1988 14:26:16
               -----------------------------------------------------------
               | NAVAL POSTGRADUATE SCHOOL           MONTEREY, CALIFORNIA |
               |---------------------------------------------------------|
               |                   ECE-4440 WORKSPACE                    |
               |                                                         |
               |                       THESIS BY                         |
               |                    Y.KATZIR,I.A.F.                      |
               |            ADVISOR:PROF. C.W. THERRIEN.                 |
               | VERSION 1.0                            SEPTEMBER 1987   |
               -----------------------------------------------------------


      )FNS
BARTLETTC       BARTLETTR       CC      CC2D    CHEB    COEFF   COS     FCC
FCC2D   FFT     FFT2D   FLCV    HAMMINGC        HAMMINGR        HANNINGC
HANNINGR        IFFT    IFFT2D  ILPFILT INFORMATION     LCV     MCCLEL  MEAN
PI      PROTFILT        RECTANR SHIFT   SHIFT2D SIN     TRANSFNC        UNSHIFT
UNSHIFT2D       XCONJ   XEXPO   XMAGN   XPHAS   XSQRT   XTIME
```

```
     ∇CC2D[⎕]∇
[0]   Y←X CC2D H;I1;I2;⎕IO;N1;N2;T1;T2
[1]   ⍝
[2]   ⍝   2-D CIRCULAR CONVOLUTION
[3]   ⍝
[4]    →((∧/(ρX)=ρH)∧2=ρρX)/BEGIN
[5]    ⎕←'IMPROPER ARGUMENTS'
[6]    →0
[7]   BEGIN:⎕IO←0
[8]    N1←1↑ρX
[9]    N2←1↓ρX
[10]   Y←(N2,N1)ρ0
[11]   I1←0
[12]  LOOP1:T1←(-(I1+1))⌽⌽H
[13]   I2←0
[14]  LOOP2:T2←(-(I2+1))⊖⊖T1
[15]   Y[I2;I1]←+/+/X×T2
[16]   →(N2>I2←I2+1)/LOOP2
[17]   →(N1>I1←I1+1)/LOOP1
[18]   →0
[19]  ⍝
[20]  ⍝  THE INPUTS NEED TO BE 2-D AND THE SAME SIZE.
[21]  ⍝
[22]  ⍝  Y.KATZIR I.A.F.,SEPTEMBER 1987
[23]  ⍝
```

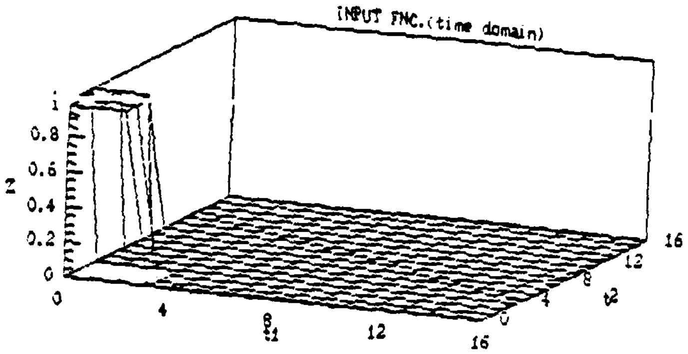
```
      X
 2 2 1 0
 1 1 0 0
 0 0 0 0
 0 0 0 0

      H
 1 1 0 0
 1 1 0 0
 1 0 0 0
 0 0 0 0

      Y←X CC2D H

      Y
 2 4 3 1
 3 6 4 1
 3 4 2 0
 1 1 0 0


      Z1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

      X CC2D Z1
IMPROPER ARGUMENTS


      Z2
 1 2
 3 4

      X CC2D Z2
IMPROPER ARGUMENTS
```

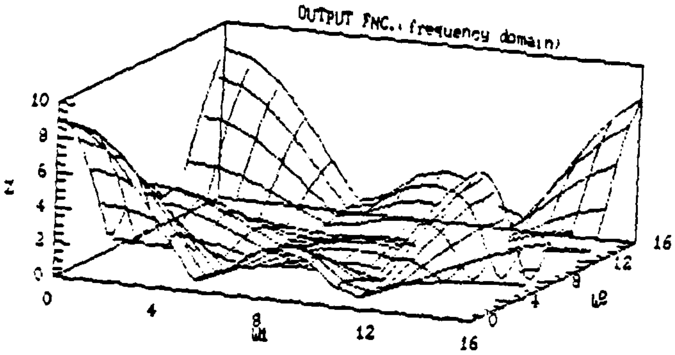
```
      ∇FFT2D[⎕]∇
[0]   Y←FFT2D X;⎕IO;NROW;NCOL;I;J;G
[1]   ⍝
[2]   ⍝  2-DIMENSIONAL FFT
[3]   ⍝
[4]    ⎕IO←0
[5]   ⍝  CONVERT REAL ARRAY TO COMPLEX ARRAY
[6]    I←⍴(⍴X)
[7]    →(I=3)/CONTIN
[8]    NROW←⍴X[;1]
[9]    NCOL←⍴X[1;]
[10]   G←(2,NROW,NCOL)⍴0
[11]   G[0;;]←X
[12]   X←G
[13]  ⍝ COMPUTATION OF FFT
[14]  CONTIN:NROW←⍴X[1;;1]
[15]   NCOL←⍴X[1;1;]
[16]   G←X×0
[17]   Y←G
[18]   I←J←0
[19]  COLLOOP:G[;;J]←FFT X[;;J]
[20]   →(NCOL>J←J+1)/COLLOOP
[21]  ROWLOOP:Y[;I;]←FFT G[;I;]
[22]   →(NROW>I←I+1)/ROWLOOP
[23]  ⍝  IF OUTPUT ARRAY IS REAL VALUE.2ND PLANE IS DELETED.
[24]   G[0;;]←Y[0;;]+Y[1;;]
[25]   G[1;;]←Y[1;;]
[26]   I←+/+/+/(G≠Y)
[27]   →(I=0)/STPNOW
[28]   →0
[29]  STPNOW:Y←(1,NROW,NCOL)↑Y
[30]   Y←(NROW,NCOL)⍴Y
[31]  ⍝
[32]  ⍝  FFT FUNCTION HAVE BEEN USED.
[33]  ⍝
[34]  ⍝  Y.KATZIR.I.A.F.,SEPTEMBER 1987
[35]  ⍝
```

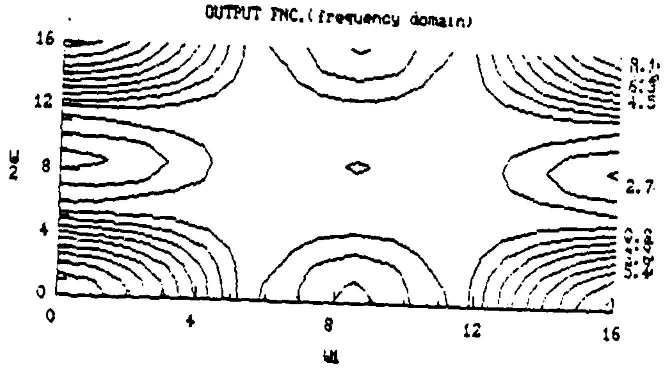
```
    ∇IFFT2D[⎕]∇
Y←IFFT2D X;⎕IO;NROW;NCOL;I;J;G
⍝
⍝  2-DIMENSIONAL INVERSE FFT
⍝
 ⎕IO←0
⍝  CONVERTING REAL ARRAY TO COMPLEX ARRAY
 I←⍴(⍴X)
 →(I=3)/CONTIN
 NROW←⍴X[;1]
 NCOL←⍴X[1;]
 G←(2,NROW,NCOL)⍴0
 G[0;;]←X
 X←G
⍝ COMPUTATION OF IFFT
CONTIN:NROW←⍴X[0;;1]
 NCOL←⍴X[0;1;]
 G←X×0
 Y←G
 I←J←0
COLLOOP:G[;;J]←IFFT X[;;J]
 →(NCOL>J←J+1)/COLLOOP
ROWLOOP:Y[;I;]←IFFT G[;I;]
 →(NROW>I←I+1)/ROWLOOP
⍝  IF OUTPUT ARRAY IS REAL VALUE,2ND PLANE IS DELETED.
 G[0;;]←Y[0;;]+Y[1;;]
 G[1;;]←Y[1;;]
 I←+/+/+/(G≠Y)
 →(I=0)/STPNOW
 →0
STPNOW:Y←(1,NROW,NCOL)↑Y
 Y←(NROW,NCOL)⍴Y
⍝
⍝  IFFT FUNCTION HAS BEEN USED.
⍝
⍝  Y.KATZIR,I.A.F.,SEPTEMBER 1987
⍝
```

```
      H1
1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

      Y←FFT2D H1

      Y
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1

      H2←IFFT2D Y

      H2
1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

      X1
2 1
1 0

1 1
1 0

      Y←FFT2D X1
      Y
 4  2
 2  0

 3  1
 1 ¯1

      X2←IFFT2D Y

      X2
2 1
1 0

1 1
1 0
```

```
      Z
1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

      Z1←FFT2D Z
      Z2←XMAGN Z1

      Z2
9     8.5   7.2   5.3   3     0.7   1.2   2.5   3     2.5   1.2
      0.7   3     5.3   7.2   8.5
8.5   8.1   6.9   5     2.8   0.67  1.2   2.4   2.8   2.4   1.2
      0.67  2.8   5     6.9   8.1
7.2   6.9   5.8   4.3   2.4   0.57  1     2     2.4   2     1
      0.57  2.4   4.3   5.8   6.9
5.3   5     4.3   3.1   1.8   0.41  0.73  1.5   1.8   1.5   0.73
      0.41  1.8   3.1   4.3   5
3     2.8   2.4   1.8   1     0.23  0.41  0.85  1     0.85  0.41
      0.23  1     1.8   2.4   2.8
0.7   0.67  0.57  0.41  0.23  0.055 0.097 0.2   0.23  0.2   0.097
      0.055 0.23  0.41  0.57  0.67
1.2   1.2   1     0.73  0.41  0.097 0.17  0.35  0.41  0.35  0.17
      0.097 0.41  0.73  1     1.2
2.5   2.4   2     1.5   0.85  0.2   0.35  0.72  0.85  0.72  0.35
      0.2   0.85  1.5   2     2.4
3     2.8   2.4   1.8   1     0.23  0.41  0.85  1     0.85  0.41
      0.23  1     1.8   2.4   2.8
2.5   2.4   2     1.5   0.85  0.2   0.35  0.72  0.85  0.72  0.35
      0.2   0.85  1.5   2     2.4
1.2   1.2   1     0.73  0.41  0.097 0.17  0.35  0.41  0.35  0.17
      0.097 0.41  0.73  1     1.2
0.7   0.67  0.57  0.41  0.23  0.055 0.097 0.2   0.23  0.2   0.097
      0.055 0.23  0.41  0.57  0.67
3     2.8   2.4   1.8   1     0.23  0.41  0.85  1     0.85  0.41
      0.23  1     1.8   2.4   2.8
5.3   5     4.3   3.1   1.8   0.41  0.73  1.5   1.8   1.5   0.73
      0.41  1.8   3.1   4.3   5
7.2   6.9   5.8   4.3   2.4   0.57  1     2     2.4   2     1
      0.57  2.4   4.3   5.8   6.9
8.5   8.1   6.9   5     2.8   0.67  1.2   2.4   2.8   2.4   1.2
      0.67  2.8   5     6.9   8.1
```


INPUT FNC.(time domain)
z
t1
t2


OUTPUT FNC.(frequency domain)
z
w1
w2


OUTPUT FNC.(frequency domain)
w2
w1
2.7

```
      ∇SHIFT[⎕]∇
[0]   Y←SHIFT X
[1]   ⍝
[2]   ⍝  SHIFTING PERIODIC DATA TO AN INTERVAL STARTING AT THE ORIGIN
[3]   ⍝
[4]    Y←(⌊(¯1+⍴X)÷2)⌽X
[5]    →0
[6]   ⍝
[7]   ⍝  SOURCE:M-D SIGNAL PROCESSING W.S.
[8]   ⍝

      X
0 0 0 0 1 0 0 0 0

      X1←SHIFT X

      X1
1 0 0 0 0 0 0 0 0


      ∇UNSHIFT[⎕]∇
[0]   Y←UNSHIFT X
[1]   ⍝
[2]   ⍝  SHIFTING PERIODIC DATA TO AN INTERVAL CENTERED AT THE ORIGIN
[3]   ⍝
[4]    Y←(-⌊(¯1+⍴X)÷2)⌽X
[5]    →0
[6]   ⍝
[7]   ⍝  SOURCE:M-D SIGNAL PROCESSING W.S.
[8]   ⍝


      X1
1 0 0 0 0 0 0 0 0

      X2←UNSHIFT X1

      X2
0 0 0 0 1 0 0 0 0
```

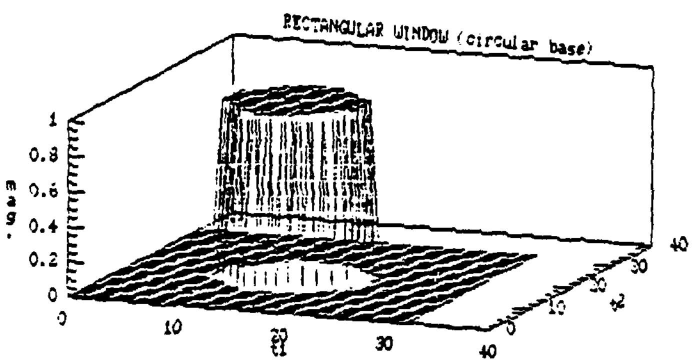
```
      ∇SHIFT2D[⎕]∇
[0]   Y←SHIFT2D X
[1]   ⍝
[2]   ⍝  SHIFTING 2-D PERIODIC DATA TO AN INTERVAL STARTING AT THE ORIGIN
[3]   ⍝
[4]    Y←(⌊(¯1+1↑1↓⍴X)÷2)⌽(⌊(¯1+1↑⍴X)÷2)⊖X
[5]    →0
[6]   ⍝
[7]   ⍝  SOURCE:M-D SIGNAL PROCESSING W.S.
[8]   ⍝
```

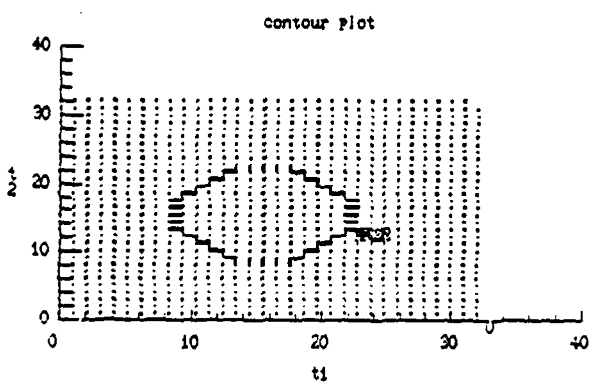
```
      ∇UNSHIFT2D[⎕]∇
[0]   Y←UNSHIFT2D X
[1]   ⍝
[2]   ⍝  SHIFTING 2-D PERIODIC DATA TO AN INTERVAL CENTERED AT THE ORIGIN
[3]   ⍝
[4]    Y←(-⌊(¯1+1↑1↓⍴X)÷2)⌽(-⌊(¯1+1↑⍴X)÷2)⊖X
[5]    →0
[6]   ⍝
[7]   ⍝  SOURCE:M-D SIGNAL PROCESSING W.S.
[8]   ⍝
```

```
      Z
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0

      Z1←SHIFT2D Z

      Z1
1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0

      Z2←UNSHIFT2D Z1

      Z2
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
```

```
     ∇ILPFILT[⎕]∇
[0]   W←R ILPFILT P;⎕IO;I;J;RR;RB;PB
[1]   ⍝
[2]   ⍝  IDEAL LP FILTER (CIRCULAR BASE) FREQUENCY RESPONSE
[3]   ⍝
[4]    ⎕IO←0
[5]    W←(P,P)⍴0
[6]    I←J←0
[7]    PB←(P÷2)-1
[8]    RB←(P×R)÷(2×PI)
[9]   LOOP:RR←(((I-PB)*2)+((J-PB)*2))*0.5
[10]   →(RR>RB)/ZERO
[11]   W[I;J]←1
[12]  ZERO:→(P>J←J+1)/LOOP
[13]   J←0
[14]   →(P>I←I+1)/LOOP
[15]   →0
[16]  ⍝
[17]  ⍝  R←THE RADIUS OF THE BASE (GAIN=1)
[18]  ⍝  P←THE DIMENSION OF THE MATRIX
[19]  ⍝  W:THE IDEAL FREQUENCY RESPONSE
[20]  ⍝
[21]  ⍝  Y.KATZIR,I.A.F., OCTOBER 1987
[22]  ⍝
```

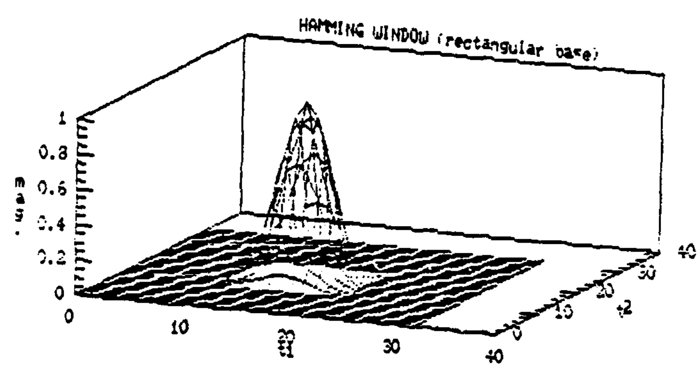
```
II←0.4×PI
N←32

WC←II ILPFILT N

0 ⎕POKE 116
```

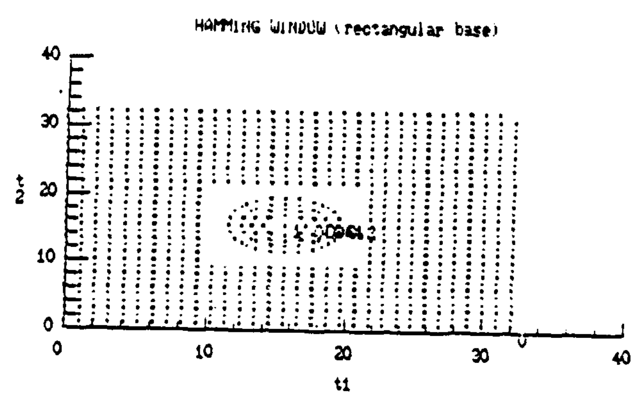




```
     ∇HAMMINGR[⎕]∇
  WR←II HAMMINGR N;⎕IO;HW1;HW2;J;I;ROW;COL;S
  ⍝
  ⍝  RECTANGULAR BASE HAMMING WINDOW
  ⍝
   ⎕IO←0
   WR←(N,N)⍴0
   HW1←HW2←II⍴0
   S←(II-1)÷2
   ROW←0
  ROWLOOP:COL←0
   I←ROW+(2×S)
   HW1[ROW]←0.54+(0.46×COS((PI×(ROW-S))÷S))
  COLLOOP:J←COL+(2×S)
   HW2[COL]←0.54+(0.46×COS((PI×(COL-S))÷S))
   WR[I;J]←HW1[ROW]×HW2[COL]
   →(II>COL←COL+1)/COLLOOP
   →(II>ROW←ROW+1)/ROWLOOP
   →0
  ⍝
  ⍝  II←WINDOW BASE DIMENSION
  ⍝  N←MATRIX DIMENSION
  ⍝  WR:THE WINDOW MATRIX
  ⍝
  ⍝  Y.KATZIR,I.A.F.,SEPTEMBER 1987
  ⍝
```

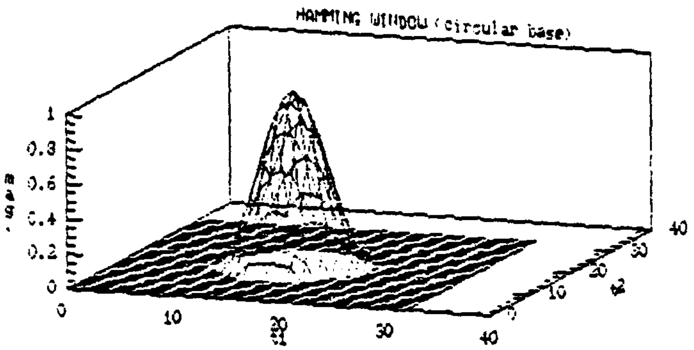
```
II←11
N←32

WR←II HAMMING2 N

0 ⎕POKE 116
```

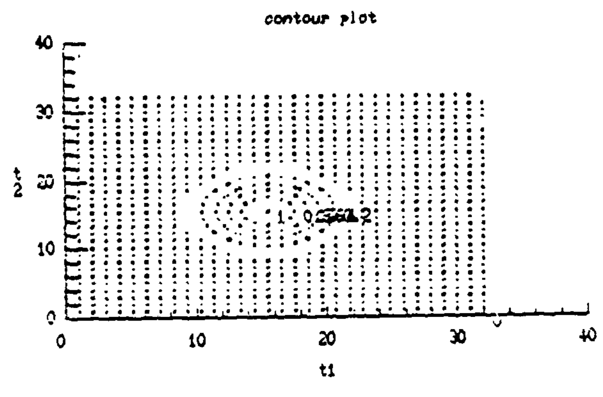




```
     ∇HAMMINGC[⎕]∇
    WC←II HAMMINGC N;⎕IO;I;J;PR;RB;RR
    ⍝
    ⍝  CIRCULAR BASE HAMMING WINDOW
    ⍝
     ⎕IO←0
     WC←(N,N)⍴0
     I←J←0
     PR←(N÷2)-1
     RB←(II×N)÷(2×PI)
    LOOP:RR←(((I-PR)*2)+((J-PR)*2))*0.5
     →(RB<RR)/CONT
     WC[J;I]←0.54+(0.46×COS((PI×RR)÷RB))
    CONT:→(N>J←J+1)/LOOP
     J←0
     →(N>I←I+1)/LOOP
     →0
    ⍝
    ⍝  II←THE RADIUS OF THE BASE
    ⍝  N←THE DIMENSION OF THE MATRIX
    ⍝  WC:THE HAMMING WINDOW
    ⍝
    ⍝  Y.KATZIR .I.A.F., OCTOBER 1987
    ⍝
```

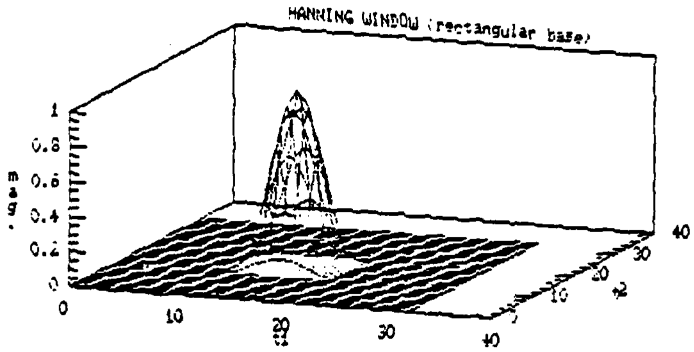
```
II←0.4×PI
N←32

WC←II HAMMINGC N

0 ⎕POKE 116
```

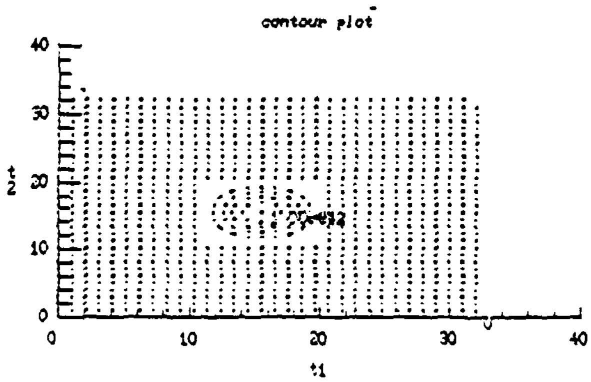




```
    ∇HANNINGR[⎕]G
[0]  WR←II HANNINGR N;⎕IO;HW1;HW2;J;I;ROW;COL;S
[1]  ⍝
[2]  ⍝  RECTANGULAR BASE HANNING WINDOW
[3]  ⍝
[4]   ⎕IO←0
[5]   WR←(N,N)⍴0
[6]   HW1←HW2←II⍴0
[7]   S←(II-1)÷2
[8]   ROW←0
[9]  ROWLOOP:COL←0
[10]  I←ROW+(2×S)
[11]  HW1[ROW]←0.5×(1+(COS((PI×(ROW-S))÷S)))
[12] COLLOOP:J←COL+(2×S)
[13]  HW2[COL]←0.5×(1+(COS((PI×(COL-S))÷S)))
[14]  WR[I;J]←HW1[ROW]×HW2[COL]
[15]  →(II>COL←COL+1)/COLLOOP
[16]  →(II>ROW←ROW+1)/ROWLOOP
[17]  →0
[18] ⍝
[19] ⍝  II←WINDOW BASE DIMENSION
[20] ⍝  N←MATRIX DIMENSION
[21] ⍝  WR:THE WINDOW MATRIX
[22] ⍝
[23] ⍝  Y.KATZIR,I.A.F.,SEPTEMBER 1987
[24] ⍝
```

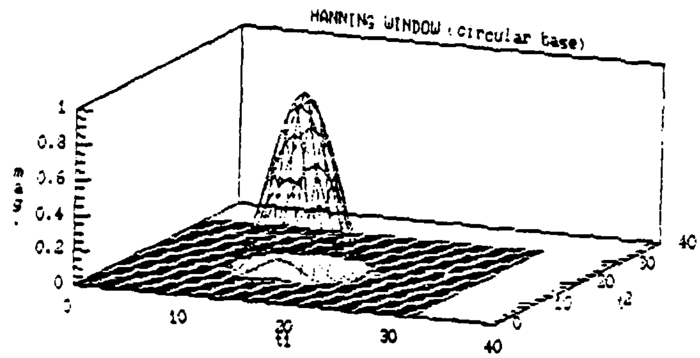
```
II←11
N←32

WR←II HANNING2 N

0 ⎕POKE 116
```

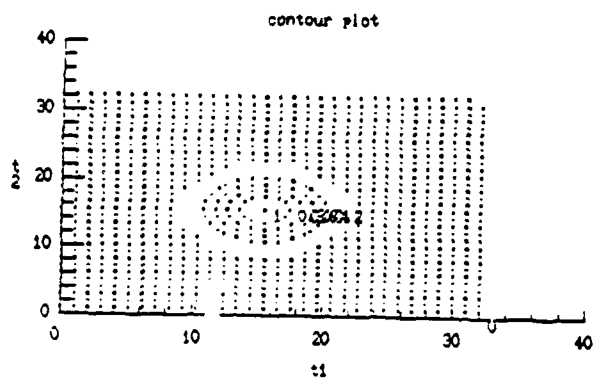




```
     ∇HANNINGC[⎕]∇
    WC←II HANNINGC N;⎕IO;I;J;PR;RB;RR
    ⍝
    ⍝  CIRCULAR BASE HANNING WINDOW
    ⍝
     ⎕IO←0
     WC←(N,N)⍴0
     I←J←0
     PR←(N÷2)-1
     RB←(II×N)÷(2×PI)
    LOOP:RR←(((I-PR)*2)+((J-PR)*2))*0.5
     →(RB<RR)/CONT
     WC[J;I]←0.5×(1+(COS((PI×RR)÷RB)))
    CONT:→(N>J←J+1)/LOOP
     J←0
     →(N>I←I+1)/LOOP
     →0
    ⍝
    ⍝  II←THE RADIUS OF THE BASE
    ⍝  N←THE DIMENSION OF THE MATRIX
    ⍝  WC:THE HANNING WINDOW
    ⍝
    ⍝  Y.KATZIR,I.A.F., OCTOBER 1987
    ⍝
```

```
II←0.4×PI
N←32

WC←II HANNINGC N

0 ⎕POKE 116
```



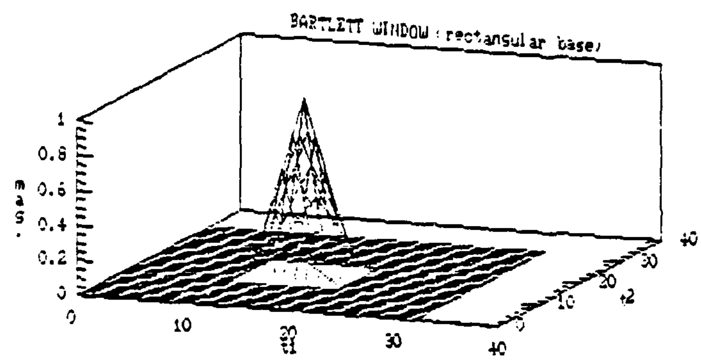


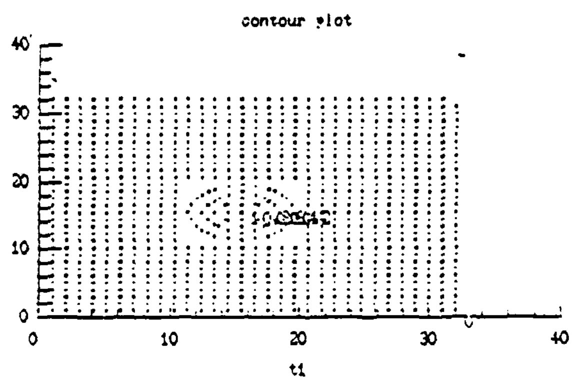
```
      ∇BARTLETTR[⎕]∇
[0]   WR←II BARTLETTR N;⎕IO;HW1;HW2;J;I;ROW;COL;S
[1]   ⍝
[2]   ⍝  RECTANGULAR BASE BARTLETT WINDOW
[3]   ⍝
[4]    ⎕IO←0
[5]    WR←(N,N)ρ0
[6]    HW1←HW2←IIρ0
[7]    S←(II-1)÷2
[8]    ROW←0
[9]   ROWLOOP:COL←0
[10]   I←ROW+(2×S)
[11]   HW1[ROW]←1-(|(ROW-S))÷S
[12]  COLLOOP:J←COL+(2×S)
[13]   HW2[COL]←1-(|(COL-S))÷S
[14]   WR[I;J]←HW1[ROW]×HW2[COL]
[15]   →(II>COL←COL+1)/COLLOOP
[16]   →(II>ROW←ROW+1)/ROWLOOP
[17]   →0
[18]  ⍝
[19]  ⍝  II←WINDOW BASE DIMENSION
[20]  ⍝  N←MATRIX DIMENSION
[21]  ⍝  WR:THE WINDOW MATRIX
[22]  ⍝
[23]  ⍝  Y.KATZIR.I.A.F.,SEPTEMBER 1987
[24]  ⍝
```

II←11
N←32

WR←II BARTLETTR N

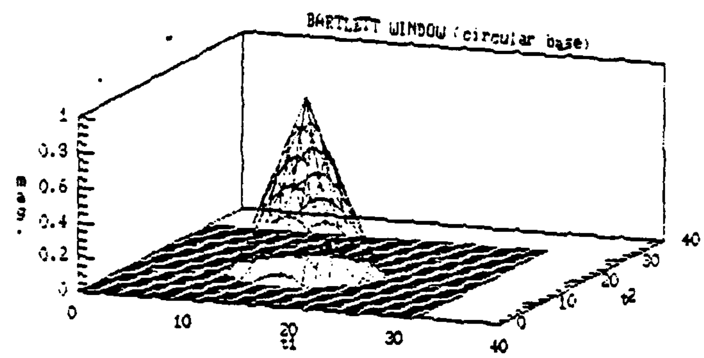
0 0POKE 116

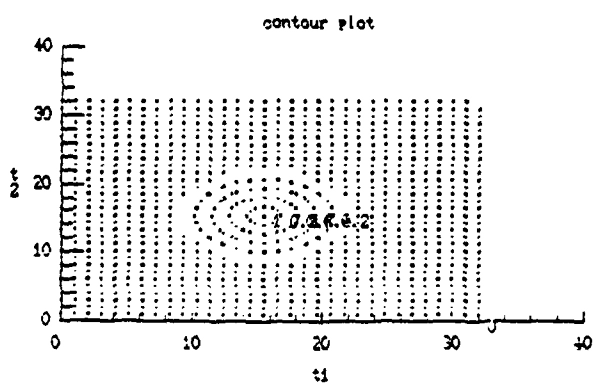




```
      ∇BARTLETTC[⎕]∇
[0]   WC←II BARTLETTC N;⎕IO;I;J;PR;RB;RR
[1]   ⍝
[2]   ⍝  CIRCULAR BASE BARTLETT WINDOW
[3]   ⍝
[4]    ⎕IO←0
[5]    WC←(N,N)⍴0
[6]    I←J←0
[7]    PR←(N÷2)-1
[8]    RB←(II×N)÷(2×PI)
[9]   LOOP:RR←(((I-PR)*2)+((J-PR)*2))*0.5
[10]   →(RB<RR)/CONT
[11]   WC[J;I]←1-(RR÷RB)
[12]  CONT:→(N>J←J+1)/LOOP
[13]   J←0
[14]   →(N>I←I+1)/LOOP
[15]   →0
[16]  ⍝
[17]  ⍝  II←THE RADIUS OF THE BASE
[18]  ⍝  N←THE DIMENSION OF THE MATRIX
[19]  ⍝  WC:THE HAMMING WINDOW
[20]  ⍝
[21]  ⍝  Y.KATZIR ,I.A.F., OCTOBER 1987
[22]  ⍝
```

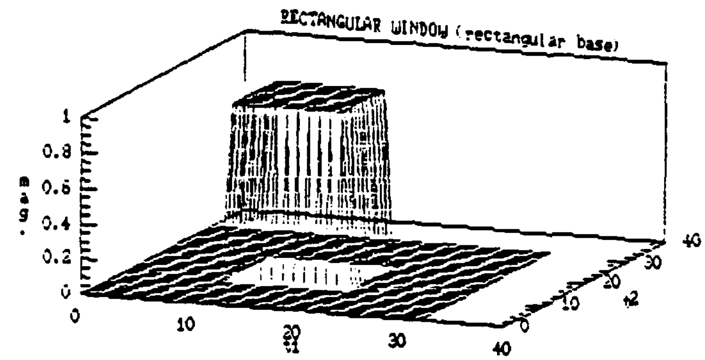
```
II←0.4×PI
N←32

WC←II BARTLETTC N

0 ⎕POKE 116
```

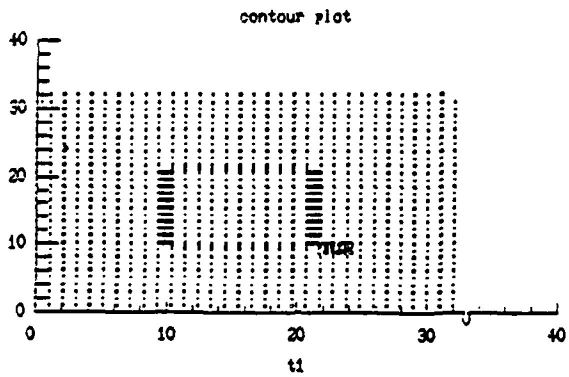




```
      ∇RECTANR[⎕]∇
[0]   WR←II RECTANR N;⎕IO;HW1;HW2;J;I;ROW;COL;S
[1]   ⍝
[2]   ⍝  RECTANGULAR BASE RECTANGULAR WINDOW
[3]   ⍝
[4]    ⎕IO←0
[5]    WR←(N,N)ρ0
[6]    HW1←HW2←IIρ0
[7]    S←(II-1)÷2
[8]    ROW←0
[9]   ROWLOOP:COL←0
[10]   I←ROW+(2×S)
[11]   HW1[ROW]←1
[12]  COLLOOP:J←COL+(2×S)
[13]   HW2[COL]←1
[14]   WR[I;J]←HW1[ROW]×HW2[COL]
[15]   →(II>COL←COL+1)/COLLOOP
[16]   →(II>ROW←ROW+1)/ROWLOOP
[17]   →0
[18]  ⍝
[19]  ⍝  II←WINDOW BASE DIMENSION
[20]  ⍝  N←MATRIX DIMENSION
[21]  ⍝  WR:THE WINDOW MATRIX
[22]  ⍝
[23]  ⍝  Y.KATZIR,I.A.F.,SEPTEMBER 1987
[24]  ⍝
```

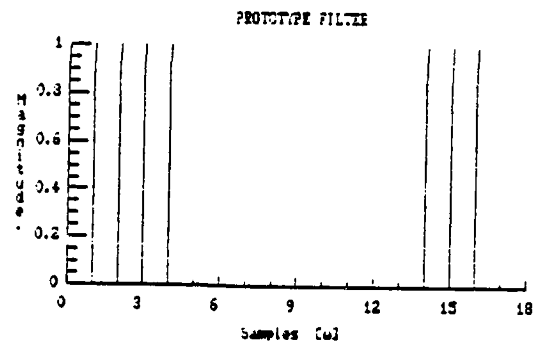
```
II←11
N←32

WR←II RECTANR N

0 DPOKE 116
```





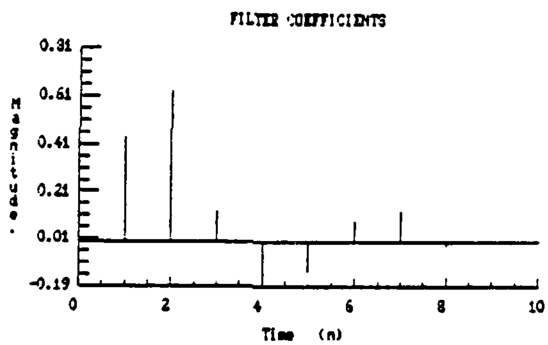
```
      ∇PROTFILT[⎕]∇
[0]   HPROT←PROTFILT SIZE;⎕IO;W;N1;N2;N
[1]   ⍝
[2]   ⍝  PRODUCING A LOWPASS PROTOTYPE FILTER
[3]   ⍝
[4]    ⎕IO←N←0
[5]    N1←SIZE+2
[6]    N2←N1-1
[7]    HPROT←SIZE⍴0
[8]   LOOP:W←((N-N2)×PI)+N1
[9]    W←|W
[10]   →(W>(0.4×PI))/ZERO
[11]   HPROT[N]←1
[12]  ZERO:→(SIZE>N←N+1)/LOOP
[13]   HPROT←SHIFT HPROT
[14]   →0
[15]  ⍝
[16]  ⍝  SIZE←THE SIZE OF THE PROTOTYPE FILTER (NO. OF SAMPLES)
[17]  ⍝  THE SHIFT FUNCTION HAS BEEN USED.
[18]  ⍝
[19]  ⍝  Y.KATZIR.I.A.F., OCTOBER 1987
[20]  ⍝
```

```
SIZE←16

HP←PROTFILT SIZE

HP
1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1


0 ⎕POKE 116
```



```
     ∇COEFF[⎕]∇
[0]   a←COEFF Hp;⎕IO;SIZE;HH;H
[1]   ⍝
[2]   ⍝  COMPUTING THE COEFFICIENTS FROM THE PROTOTYPE FILTER (USING IFFT)
[3]   ⍝
[4]    ⎕IO←0
[5]    SIZE←(ρHp)÷2
[6]    a←SIZEρ0
[7]    H←IFFT Hp
[8]    HH←(1,SIZE)↑H
[9]    HH←SIZEρHH
[10]  a[ιSIZE]←2×HH[ιSIZE]
[11]  a[0]←a[0]÷2
[12]  →0
[13] ⍝
[14] ⍝  Hp←THE PROTOTYPE SAMPLES
[15] ⍝  a:THE IMPULSE RESPONSE COEFFICIENTS
[16] ⍝  THE IFFT FUNCTION HAS BEEN USED.
[17] ⍝
[18] ⍝  Y.KATZIR,I.A.F., OCTOBER 1987
[19] ⍝
```

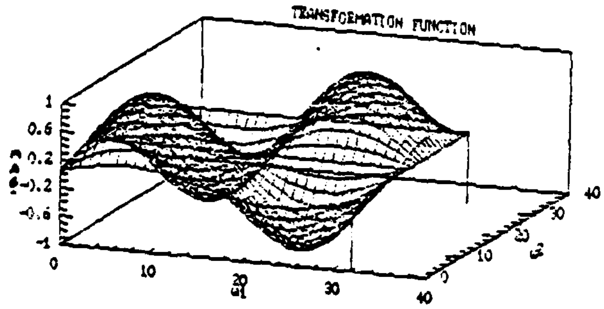
```
0
      HP
1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1

      A←COEFF HP


      A
0.438 0.628 0.125 ¯0.187 ¯0.125 0.0835 0.125 ¯0.0249

      0 ⎕POKE 116
```

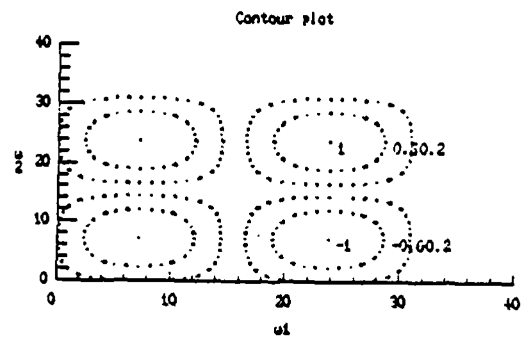


```
      ∇TRANSFNC[⎕]∇
[0]   Fww←TRANSFNC SIZE;⎕IO;A;B;ROW;COL;W1;W2;N1;N2
[1]   ⍝
[2]   ⍝  PRODUCING 2-D TRANSFORMATION FUNCTION (EXAMPLE)
[3]   ⍝
[4]    ⎕IO←0
[5]    Fww←(SIZE,SIZE)⍴0
[6]    ROW←0
[7]    N1←SIZE÷2
[8]    N2←N1-1
[9]   ROWLOOP:W1←((ROW-N2)×PI)÷N1
[10]   A←COS(W1)
[11]   COL←0
[12]  COLLOOP:W2←((COL-N2)×PI)÷N1
[13]   B←COS(W2)
[14]   Fww[ROW;COL]←((-1)+A+B+(A×B))÷2
[15]   →(SIZE>COL←COL+1)/COLLOOP
[16]   →(SIZE>ROW←ROW+1)/ROWLOOP
[17]   →0
[18]  ⍝
[19]  ⍝  THE FUNCTION PRODUCING THE TRANSFORMATION MATRIX IN ORDER TO
[20]  ⍝  DESIGN FIR FILTER USING McCLELLAN TRANSFORMATION
[21]  ⍝  SIZE←NO. OF SAMPLES IN ω1.(ω2 WILL BE THE SAME SIZE AUTOMATICALLY)
[22]  ⍝  Fww:THE TRANSFORMATION MATRIX
[23]  ⍝  IN ORDER TO CHANGE  F(ω1,ω2),REPLACE LINES [10],[13], AND [14]
[24]  ⍝  WITH THE NEW CHOICES.
[25]  ⍝
[26]  ⍝  Y.KATZIR,I.A.F., OCTOBER 1987
[27]  ⍝
```

```
      ∇TRANSFNC[⎕]∇
[0]   Fww←TRANSFNC SIZE;⎕IO;A;B;ROW,COL;W1;W2;N1;N2
[1]   ⍝
[2]   ⍝  PRODUCING 2-D TRANSFORMATION FUNCTION (EXAMPLE)
[3]   ⍝
[4]    ⎕IO←0
[5]    Fww←(SIZE,SIZE)⍴0
[6]    ROW←0
[7]    N1←SIZE÷2
[8]    N2←N1-1
[9]   ROWLOOP:W1←((ROW-N2)×PI)÷N1
[10]   A←SIN(W1)
[11]   COL←0
[12]  COLLOOP:W2←((COL-N2)×PI)÷N1
[13]   B←SIN(W2)
[14]   Fww[ROW;COL]←A×B
[15]   →(SIZE>COL←COL+1)/COLLOOP
[16]   →(SIZE>ROW←ROW+1)/ROWLOOP
[17]   →0
[18]  ⍝
[19]  ⍝  THE FUNCTION PRODUCING THE TRANSFORMATION MATRIX IN ORDER TO
[20]  ⍝  DESIGN FIR FILTER USING McCLELLAN TRANSFORMATION
[21]  ⍝  SIZE←NO. OF SAMPLES IN ω1.(ω2 WILL BE THE SAME SIZE AUTOMATICALLY)
[22]  ⍝  Fww:THE TRANSFORMATION MATRIX
[23]  ⍝  IN ORDER TO CHANGE  F(ω1,ω2),REPLACE LINES [10],[13], AND [14]
[24]  ⍝  WITH THE NEW CHOICES.
[25]  ⍝
[26]  ⍝  Y.KATZIR,I.A.F., OCTOBER 1987
[27]  ⍝
```

```
F←TRANSFNC 32

0 GPOKE 116
```





```
     ∇CHEB[⎕]∇
[0]   Y←N CHEB X
[1]   ⍝
[2]   ⍝  EVALUATING CHEBYSHEV POLYNOMIALS
[3]   ⍝
[4]    →((N=0),N=1)/A,B
[5]    Y←(2×X×(N-1)CHEB X)-(N-2)CHEB X
[6]    →0
[7]   A:Y←(⍴X)⍴1
[8]    →0
[9]   B:Y←X
[10]   →0
[11] ⍝
[12] ⍝  N←THE POLYNOMIALS ORDER
[13] ⍝  X←THE FUNCTION TO BE EVALUATED
[14] ⍝
[15] ⍝  SOURCE:MDSP W.S.
[16] ⍝
```

```
      X
1 2 3

      0 CHEB X
1 1 1
      1 CHEB X
1 2 3
      2 CHEB X
1 7 17
      3 CHEB X
1 26 99
      4 CHEB X
1 97 577
      5 CHEB X
1 362 3363
      6 CHEB X
1 1351 19601
```

```
 ∇MCCLEL[⎕]∇
Hww←a MCCLEL F;⎕IO;N;INTER;NUMBER;INTER1;INTER0
⍝
⍝  DESIGNING 2-D FILTER USING MEcCLELLAN TRANSFORMATION
⍝
 ⎕IO←0
 INTER0←INTER1←INTER←Hww←(⍴F)⍴0
 NUMBER←⍴a
 N←0
LOOP:→((N=0),N=1)/A,B
 INTER←(2×F×INTER1)-INTER0
 →C
A:INTER←(⍴F)⍴1
 INTER1←INTER
 →C
B:INTER←F
C:⎕←'CALCULATED CHEB. POLYNOM OF ORDER:',⍕N
 Hww←Hww+(a[N]×INTER)
 INTER0←INTER1
 INTER1←INTER
 →(NUMBER>N←N+1)/LOOP
 →0
⍝
⍝  F←THE TRANSFORMATION MATRIX (CALCULATED USING TRANSFORMATION
⍝  FUNCTION)
⍝  a←THE PROTOTYPE IMPULSE RESPONSE COEFFICIENTS
⍝  Hww:THE 2-D FREQUENCY RESPONSE.
⍝
⍝  Y.KATZIR,I.A.F., OCTOBER 1987
⍝
```

APPENDIX D
COMPUTER ASSIGNMENTS AND SOLUTIONS

This appendix provides a few samples of computer assignments and related solutions using the software package. These are in addition to the examples in Chapter III. The computer assignments in EC 3400 were taken from FIRST PRINCIPLES OF DISCRETE SYSTEMS AND DIGITAL SIGNAL PROCESSING by Professors R. D. Strum and D. E. Kirk [4]. The computer assignments in EC 3410 and EC 4440 were taken from Professor C. W. Therrien's homework assignments.

EC 3400 COMPUTER ASSIGNMENT [4]

The purpose of a bandstop filter is to remove the effects of the vibrations in the frequency range 6 Hz to 15 Hz from a control loop. The sampling frequency is fs=100 Hz.

a. Using the Fourier Series design procedure, calculate the filter coefficients with a rectangular window.

b. Plot the magnitude portion of the frequency response for a filter having thirty-one coefficients.

c. Repeat part (b) using a von-Hann window.

SOLUTION

1. By using the function DIGFREQ the digital upper and lower frequencies are calculated (Figure D.1). Figure D.2 describes the ideal frequency response to be generated.

```
      TUL←2⍴0
      TUL[1]←100DIGFREQ 15
      TUL[1]
0.94248
      TUL[1]÷PI
0.3

      TUL[2]←100 DIGFREQ 6
      TUL[2]
0.37699
      TUL[2]÷PI
0.12
```

Figure D.1. Calculating the Digital Frequency Using the Function DIGFREQ.

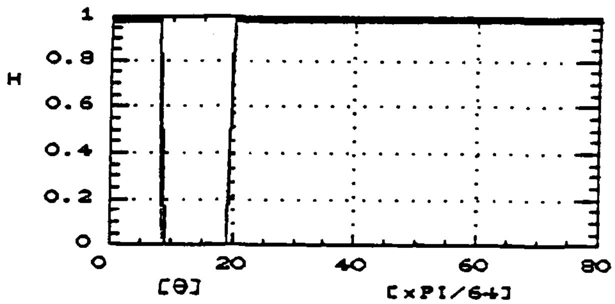


Figure D.2. The Ideal Frequency Response.

2. By using the function BSCOEFF 31 causal impulse response coefficients are calculated (See Figures D.3 and D.4).
3. The function HANNING generates 31 von-Hann window samples that are multiplied by the filter coefficients. The result is windowed coefficients (See Figures D.5 and D.6).
4. By using the function FREQRES the frequency responses with and without window are generated. The following shows the APL commands which generate the frequency responses, and Figures D.7 and D.8 show the plotted results.

```
HBS←128 FREQRES hBS
HBSW←128 FREQRES hBSW
```

```
      hBS←31 BSCOEFF TUL

      hBS
¯0.033694 ¯0.032561 ¯0.016485 ¯8.2842E¯4 ¯1.0218E¯3 ¯0.01871 ¯0.037409
      ¯0.032854 7.8548E¯3 0.07206 0.12421 0.12619 0.063217 ¯0.042416 ¯0.14
      0.82 ¯0.14034 ¯0.042416 0.063217 0.12619 0.12421 0.07206 7.8548E¯3
      ¯0.032854 ¯0.037409 ¯0.01871 ¯1.0218E¯3 ¯8.2842E¯4 ¯0.016485 ¯0.03
      ¯0.033694
```

Figure D.3. Calculating the Impulse Response Coefficients.

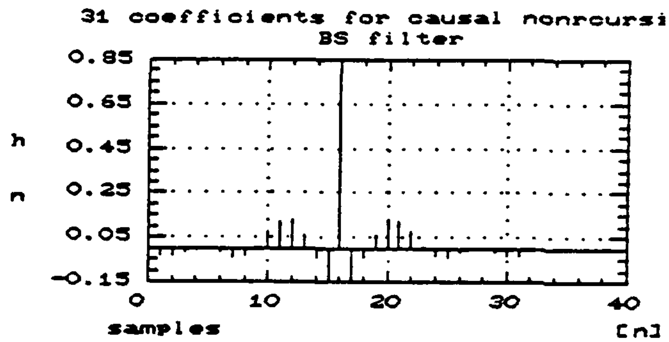


Figure D.4. The Impulse Response Coefficients.

```
      W←HANNING 31

      W
2.5653E¯3 0.02293 0.062827 0.12062 0.19395 0.2798 0.37467 0.47468 0.5757
      0.67365 0.76448 0.84448 0.91038 0.95948 0.98976 1 0.98976 0.95948
      0.91038 0.84448 0.76448 0.67365 0.57571 0.47468 0.37467 0.2798 0.194
      0.12062 0.062827 0.02293 2.5653E¯3

      hBSW←hBS×W
```

Figure D.5. Generating the Hanning Window.

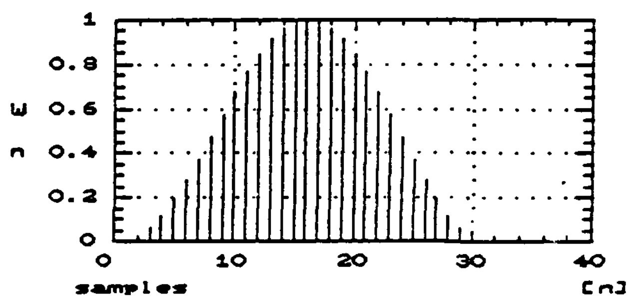


Figure D.6. The Hanning Window.

```
HBS←128 FREQRES hBS
HBSW←128 FREQRES hBSW
```

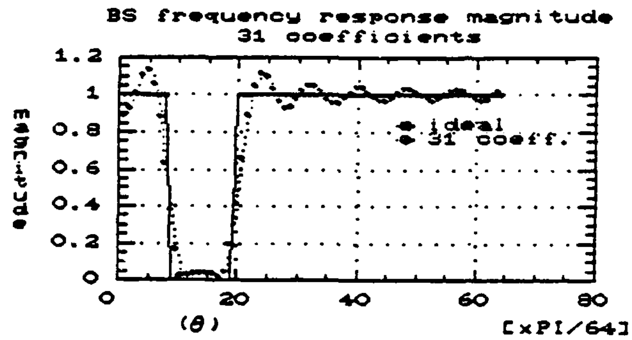


Figure D.7. The Frequency Response Magnitude.

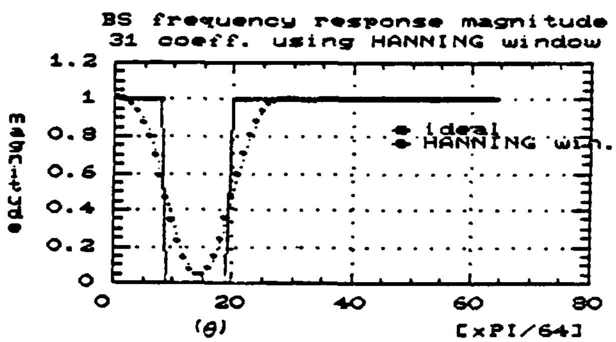
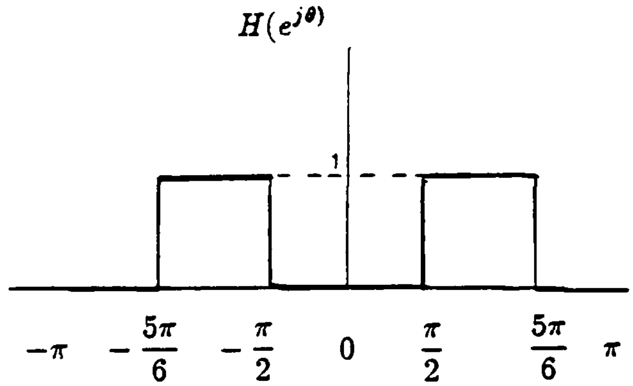


Figure D.8. The Frequency Response Magnitude with Window.

EC 3400 COMPUTER ASSIGNMENT [4]

A nonrecursive bandpass filter is to be designed using the IDFT approach. The ideal characteristic is given below:



Use 255 frequency samples and find 51 filter coefficients using:

a. A rectangular window

b. A Hamming window

c. Plot the frequency response for the filters of parts (a) and (b).

SOLUTION

1. By using the function IDEALF and choosing 256 samples, the ideal bandpass frequency response is generated (See Figure D.9).

2. By using the function FCOEFF 51 coefficients are calculated (See Figure D.10).

3. The function HAMMING generates 51 Hamming window samples that are multiplied by the filter coefficients. The result is the windowed coefficients.

```
TC←3⍴0
TC[1]←256
TC[2]←5÷6
TC[3]←0.5

H←'BP' IDEALF TC

HH←128↑H
```

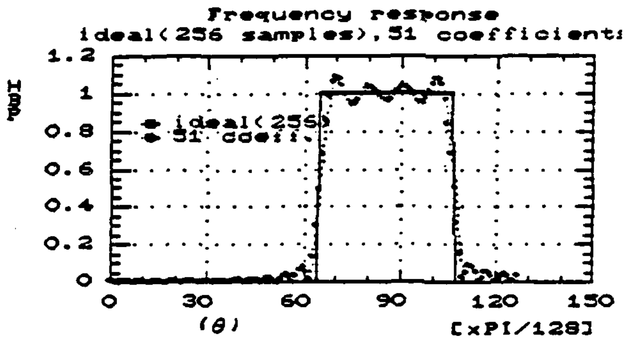
Figure D.9. Generating the Ideal Frequency Response Using 255 Samples.

```
      h←51 FCOEFF H

      h
¯6.2133E¯3 ¯4.642E¯3 2.582E¯3 0.020547 ¯0.027498 4.7036E¯3 8.7724E¯3 4.8E¯3
      ¯5.5166E¯3 ¯0.025658 0.040397 ¯0.010498 ¯0.013214 ¯5.0631E¯3 0.011
      0.036326 ¯0.069496 0.025239 0.024298 5.1718E¯3 ¯0.029088 ¯0.077918
      0.21179 ¯0.12868 ¯0.16131 0.32813 ¯0.16131 ¯0.12868 0.21179 ¯0.077
      ¯0.029088 5.1718E¯3 0.024298 0.025239 ¯0.069496 0.036326 0.011158
      ¯5.0631E¯3 ¯0.013214 ¯0.010498 0.040397 ¯0.025658 ¯5.5166E¯3 4.885E¯3
      8.7724E¯3 4.7036E¯3 ¯0.027498 0.020547 2.582E¯3 ¯4.642E¯3 ¯6.2133E¯3
```

Figure D.10. Calculating the Impulse Response Coefficients.

```
W←HAMMING 51

hW←h×W

HBP←128 FREQRES h
HBPW←128 FREQRES hW
```

Figure D.11. Generating the Hamming Window and Calculating the Frequency Response.



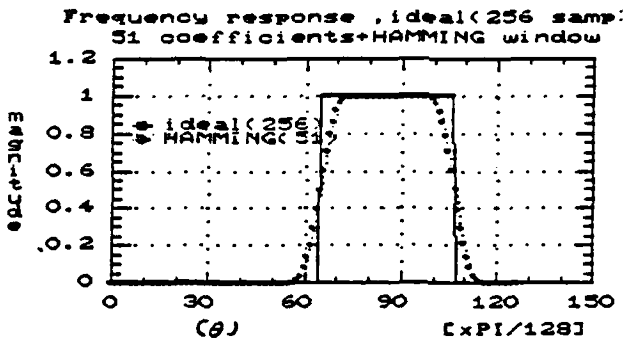


Figure D.12. The Frequency Response Plots.

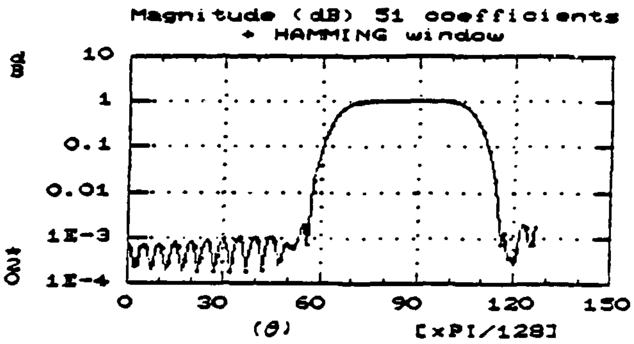


Figure D.13. The Frequency Response in dB

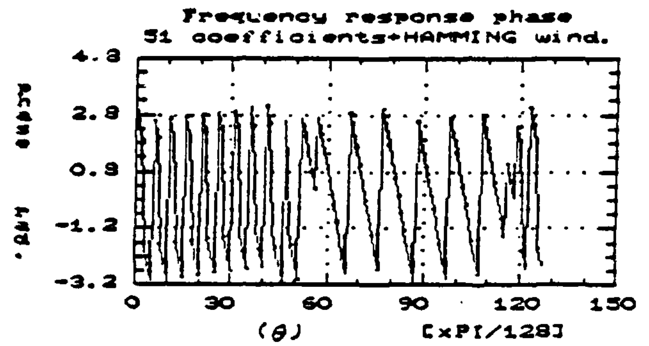


Figure D.14. The Frequency Response Phase.

EC 3410 COMPUTER ASSIGNMENT [5]

On a floppy disk you will find 4 data sets called

SL0.DAT

SL1.DAT

SL2.DAT

SL3.DAT

Each of these represents a random signal with 512 time points. The format of each file is identical. You can look at the file with an editor. The first line contains the number 512 displaying the number of points in the data set. On the remaining lines, having four numbers per line, you will find the floating point values of the signal.

1. Transfer these files to your APL WS using the UTILITY function GETDATA.
2. Plot eacn of the signals versus time and submit the plots with your assignment. Tell whether you think each of these signals seems to be more-or-less uncorrelated, positively correlated, or negatively correlated.
3. Compute the mean of each signal by averaging in time and write down your result for each signal. It is reasonable based on your plots?
4. Subtract the mean from each signal and compute and plot the sample correlation function. Attach the plots to your assignment. Was your guess about the correlation of these signals in part 2 correct?

SOLUTION

1. Using the function GETDATA and the following statements, the data sets SL0, SL1, and SL3 are transferred to the APL workspace:

```
SL0←GETDATA 'SL0.DAT'
SL1←GETDATA 'SL1.DAT
SL2←GETDATA 'SL2.DAT
SL3←GETDATA 'SL3.DAT
```

2. By using the *X-Y* plotting procedure in STATGRAPHICS the data plots are generated. (See Figures D.15, D.16, D.17, and D.18.) Another kind of plot can be obtained by using the STATGRAPHICS Time Series Procedure. The results are shown in Figures D.19, D.20, D.21, and D.22.

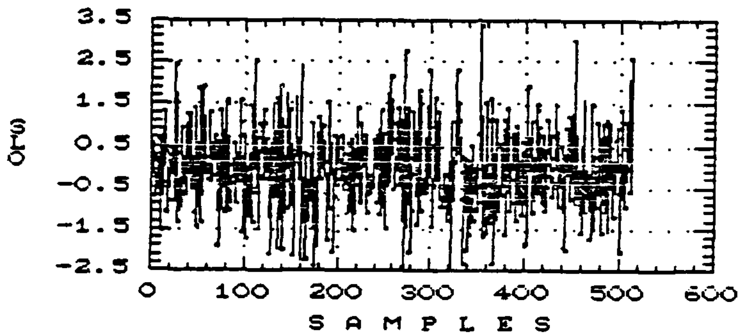


Figure D.15. SL0 Data Set.

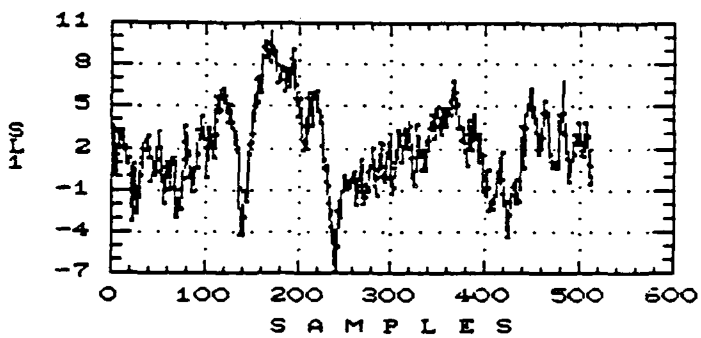


Figure D.16. SL1 Data Set.

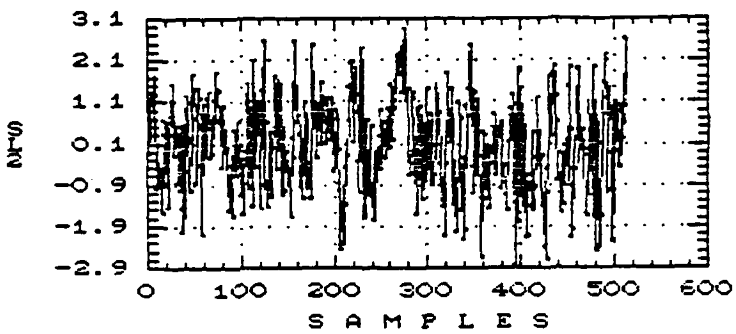


Figure D.17. SL2 Data Set.

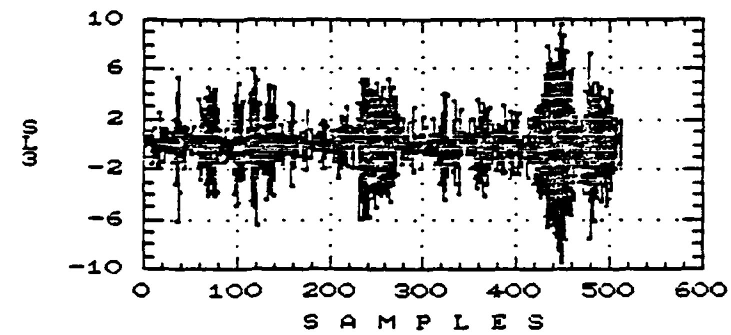


Figure D.18. SL3 Data Set.

3. From the plotting of each data set, it can be seen that:

   a. SL0 changes randomly with little consistency, this is characteristic of a white noise sequence.

   b. SL1 varies more slowly and stays stable for a while with positive values and then changes more-or-less smoothly to negative values. This indicates that it is a positively correlated data set.

   c. SL2 is very nearly like a white noise sequence.

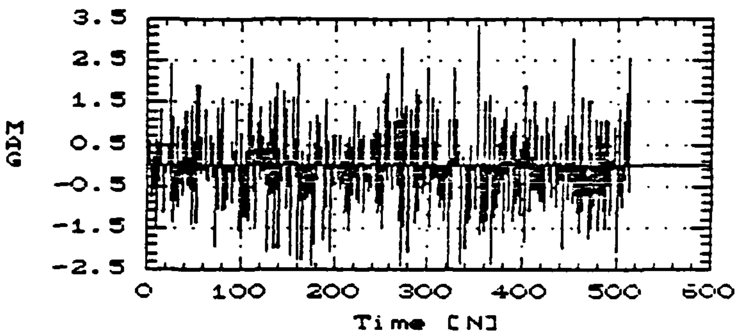


Figure D.19. SL0 Data Set Using Time Series Procedure.

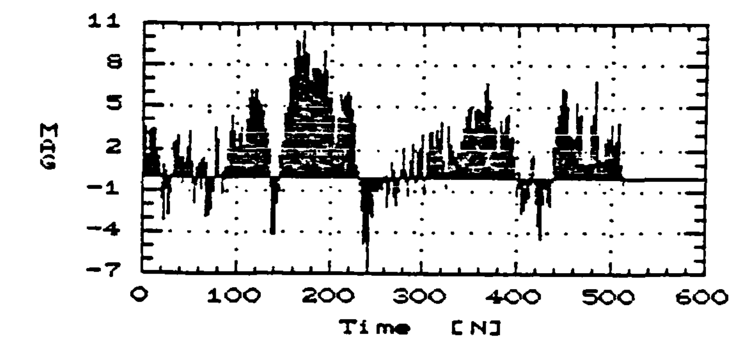


Figure D.20. SL1 Data Set Using Time Series Procedure.

d. SL3 changes at almost every step from negative to positive. This is characteristic of a negatively correlated data set.

4. Calculate the mean of each data set using the MEAN function. Then subtract the mean of each data set from the data set in order to achieve a zero mean data set. Using the following instructions, calculate the first 150 autocorrelation lags for each data set:

```
Rxx1←150 SACF SL0
Rxx2←150 SACF SL1
Rxx3←150 SACF SL2
```

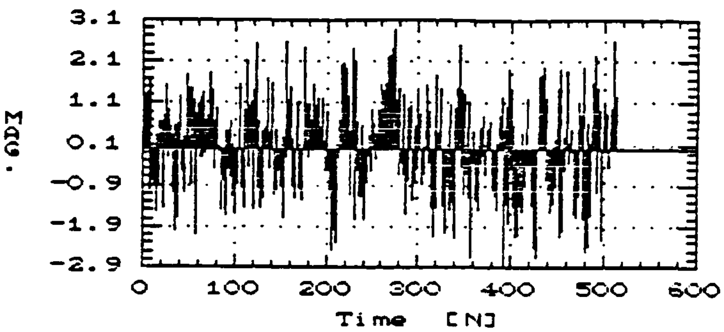


Figure D.21. SL2 Data Set Using Time Series Procedure.

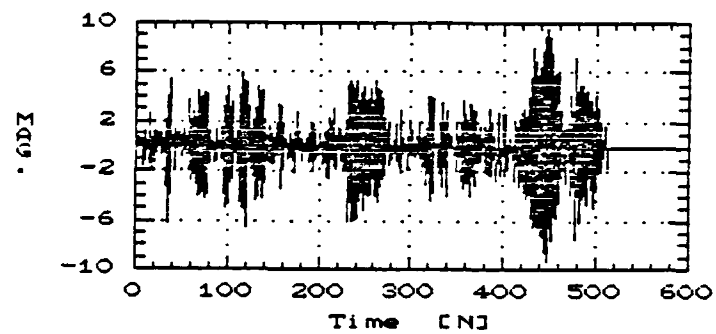


Figure D.22. SL3 Data Set Using Time Series Procedure.

```
Rxx4←150 SACF SL3
```

Using the X-Y plotting procedure in STATGRAPHICS the estimated autocorrelation is plotted as shown in Figures D.23 to D.26.

Another method for producing the autocorrelation values and plot them is to use the autocorrelation procedure in the TIME SERIES ANALYSIS section from the STATGRAPHICS Menu. The results are shown in Figures D.27 to D.30.

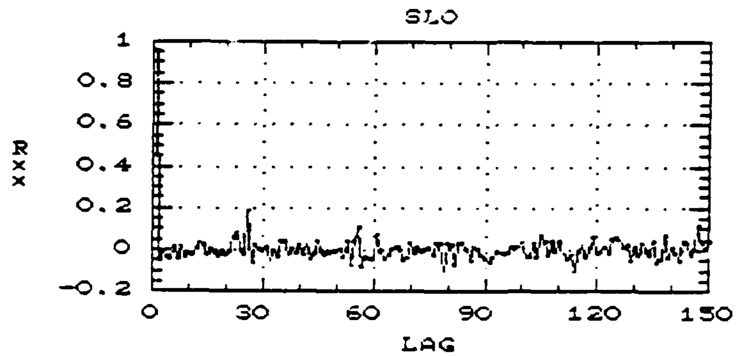


Figure D.23. Sample Autocorrelation Function of White Noise.

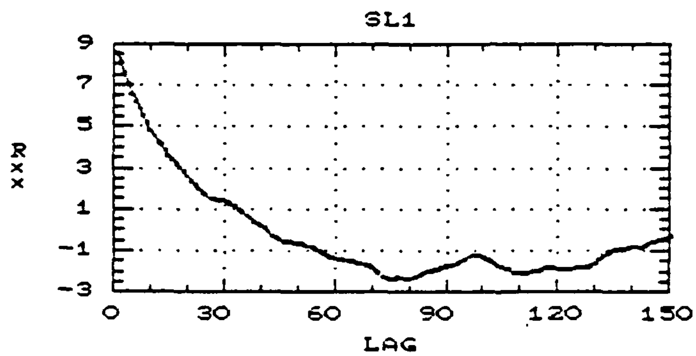


Figure D.24. Sample Autocorrelation Function of Positive Correlated Set.

Looking at the plots verifies that our characterization of the data sets in paragraph 3 was correct.

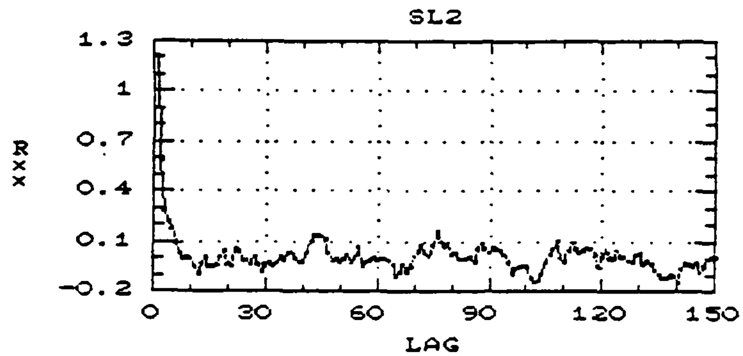


Figure D.25. Sample Autocorrelation Function of Nearly White Noise.

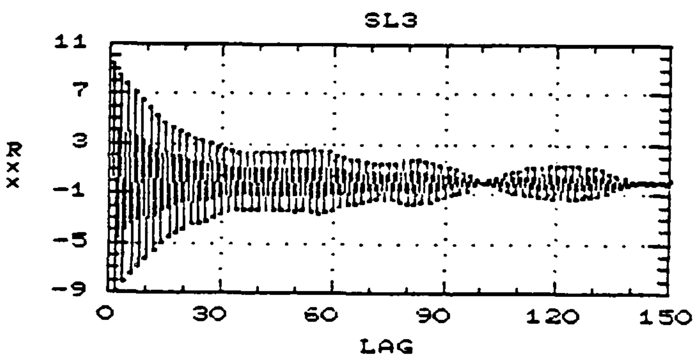


Figure D.26. Sample Autocorrelation Function of Negative Correlated Data Set.

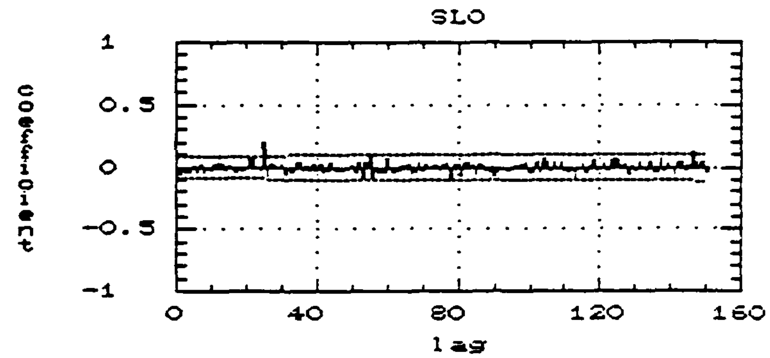


Figure D.27. Estimated Autocorrelation Using Time Series Procedure.

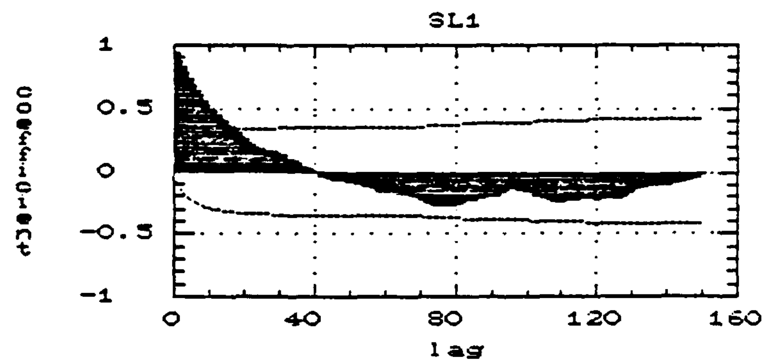


Figure D.28. Estimated Autocorrelation Using Time Series Procedure.

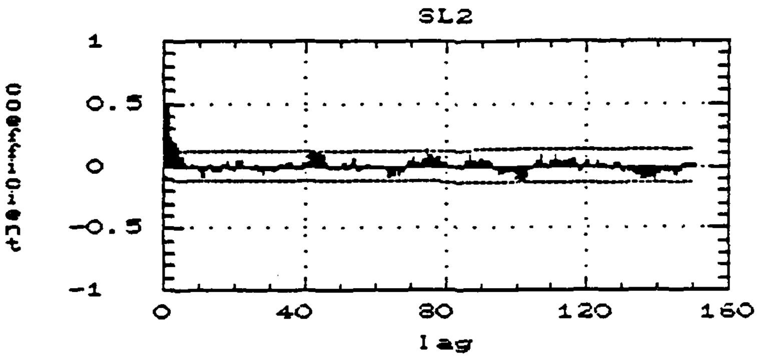


Figure D.29. Estimated Autocorrelation Using Time Series Procedure.

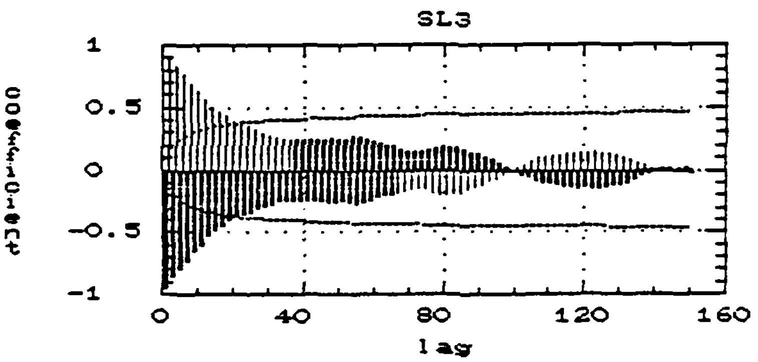


Figure D.30. Estimated Autocorrelation Using Time Series Procedure.

EC 3410 COMPUTER ASSIGNMENT [5]

Compute power spectrum estimates for the data sets SL0, SL1, SL2, and SL3 using a rectangular window. In all cases zero-pad your data appropriately before giving it to the FFT routine so that you will have computed enough points of the spectrum to give a smooth plot. (Plotting say 256 points of the spectrum should be sufficient). Assume the data has been sampled at a spacing of $T = 0.1$ ms. Label the frequency axis in hertz. Since the signals are real, it is only necessary to plot the positive frequencies.

a. Use only one 16-point segment (the first 16 points) of each data set.

b. Use 32 16-point segments.

c. Use 16 32-point segments.

d. Use 4 128-point segments.

What do you make of the results?

SOLUTION

1. The first step is to extract the first 16 samples of each data set and compute the power spectrum estimate using the function PSE. In order to obtain a smooth plot the data sets are zero padded with 240 points. (See Figure D.31.) The power spectrum estimate will have 250 points as shown on the plots of Figures D.32 to D.35.

2. By using the function PSEB (power spectrum estimation with Bartlett window), we estimate the periodogram of the SL0 data set using 30 segments of 17 points, 16 segments of 32 points, and 4 segments of 128 points as seen in Figure D.36. Figure D.37 is the periodogram plots for each case. The above process is repeated with the data sets SL1, SL2, and SL3. See Figures D.38 to D.43.

3. From these results we conclude that the first data set has a spectrum like white noise. The second and third spectra are lowpass with bandwidths of

```
      SL00←16↑SL0
      S0←240 PSE SL00
      ρS0
256
```

```
      SL11←16↑SL1
      S1←240 PSE SL11
      ρS1
256
```

```
      SL22←16↑SL2
      S2←240 PSE SL22
      ρS2
256
```

```
      SL33←12↑SL3
      S3←240 PSE SL33
      ρS3
256
```

Figure D.31. Calculating the Power Spectrum Estimates Using the Function PSE.

approximately 1 kHz and 1.2 kHz. The fourth spectrum is highpass with bandwidth of approximately 1 Khz Hz. It seems that the best estimate is obtained by using 16 segments of 32 points each.

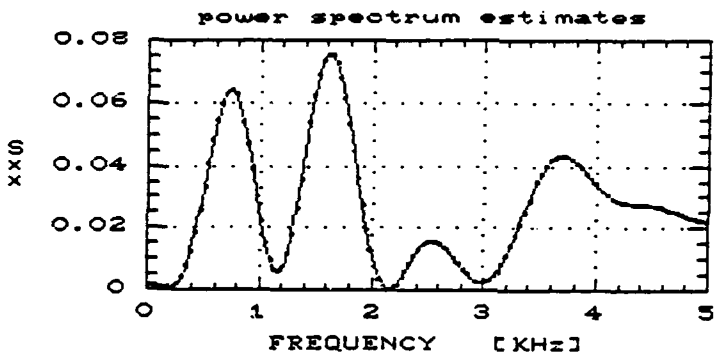


Figure D.32. The PSE of SL0 Data Set Using 16 Points.

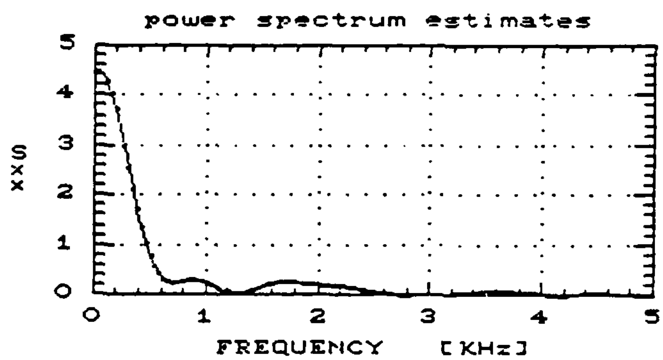


Figure D.33. The PSE of SL1 Data Set Using 16 Points.

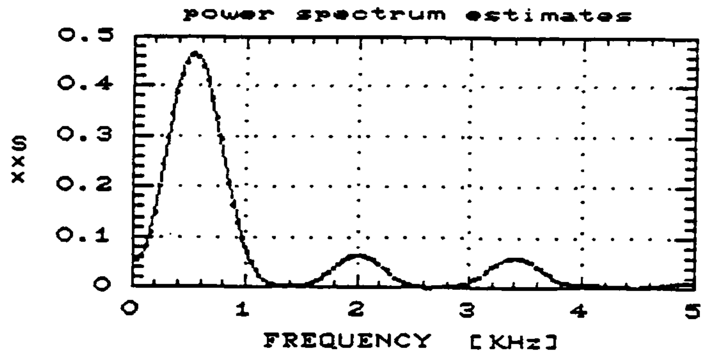


Figure D.34. The PSE of SL2 Data Set Using 16 Points.

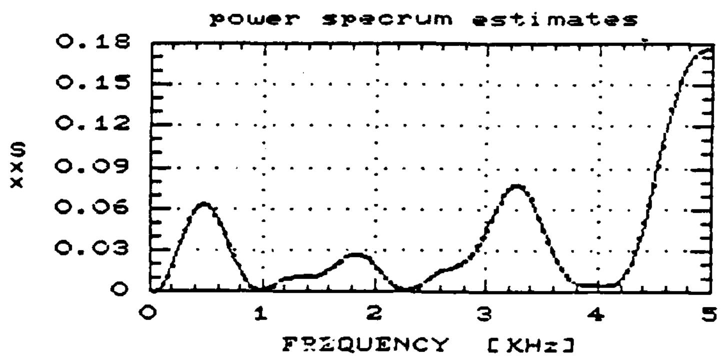


Figure D.35. The PSE of SL3 Data Set Using 16 Points.

```
SL01←17 PSEB SL0

SL01←128↑SL01
```

```
SL02←32 PSEB SL0

SL02←128↑SL02
```

```
SL03←128 PSEB SL0

SL03←128↑SL03
```

Figure D.36. Calculating Periodograms of SL0 Using Bartlett Windows.

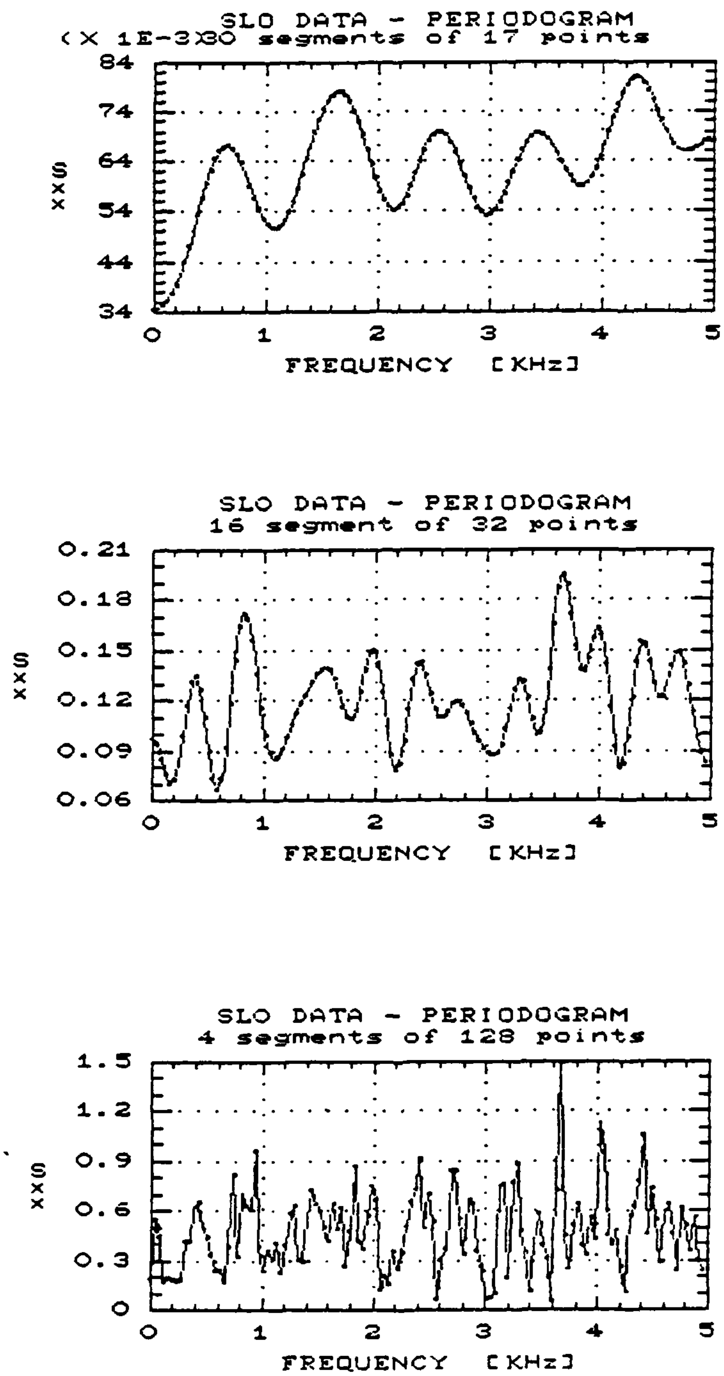


Figure D.37. Periodograms of Different Windows.

```
SL11←17 PSEB SL1
SL11←128↑SL11
```

```
SL12←32 PSEB SL1
SL12←128↑SL12
```

```
SL13←128 PSEB SL1
SL13←128↑SL13
```

Figure D.38. Calculating Periodograms of SL1 Using Bartlett Windows.

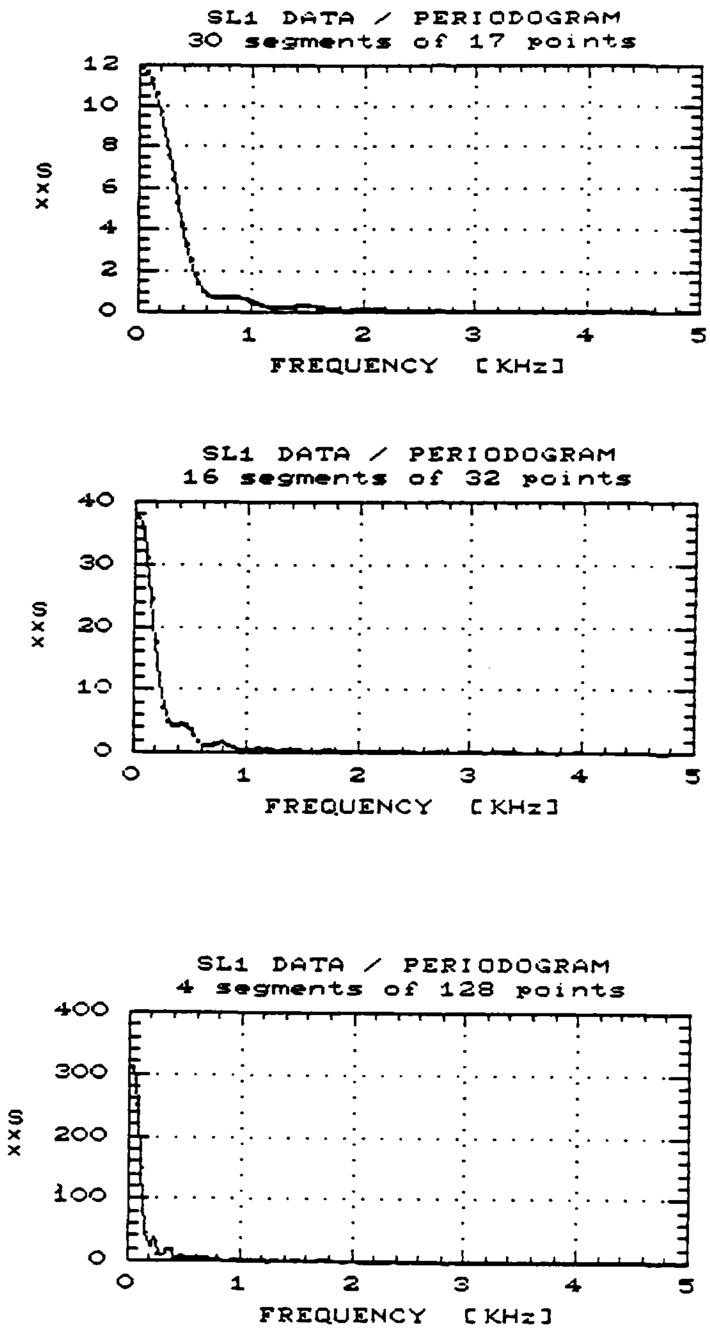


Figure D.39. Periodograms of Different Windows.

```
SL21←17 PSEB SL2
SL21←128↑SL21
```

```
SL22←32 PSEB SL2
SL22←128↑SL22
```

```
SL23←128 PSEB SL2
SL23←128↑SL23
```

Figure D.40. Calculating Periodograms of SL2 Using Bartlett Windows.

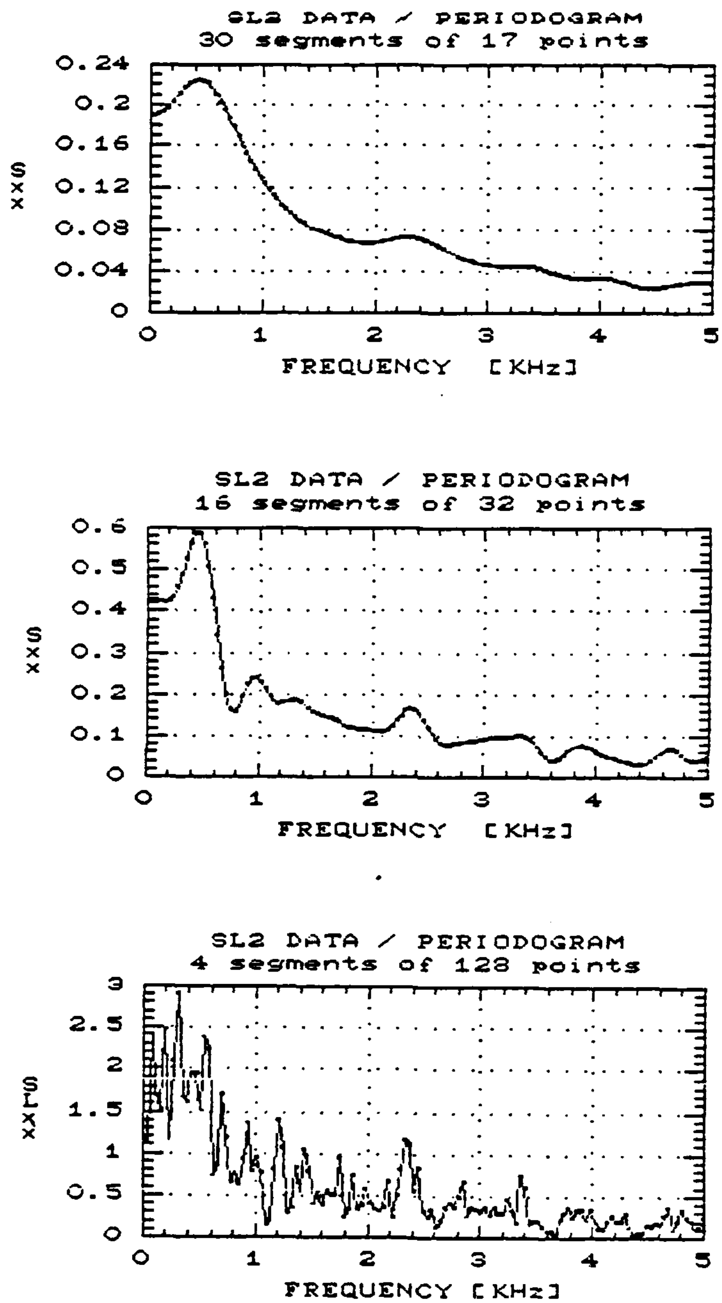


Figure D.41. Periodograms of Different Windows.

```
SL31←17 PSEB SL3
SL31←128↑SL31
```

```
SL32←32 PSEB SL3
SL32←128↑SL32
```

```
SL33←128 PSEB SL3
SL33←128↑SL33
```

Figure D.42. Calculating Periodograms of SL3 Using Bartlett Windows.

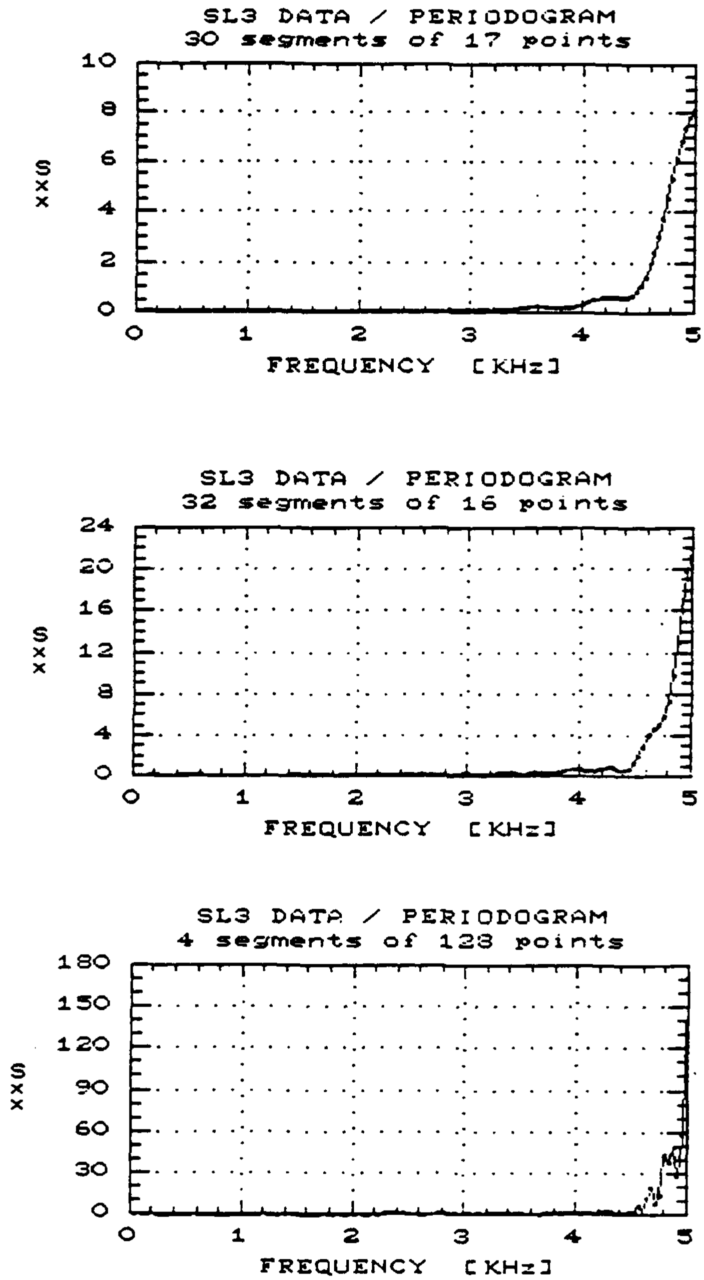


Figure D.43. Periodograms of Different Windows.

EC 4440 COMPUTER ASSIGNMENT-FILTER DESIGN [6]

This problem deals with the design of a lowpass filter by the windowing method. Use APL to do this problem. Use STATGRAPHICS to generate 3-D and contour plots.

1. Start with the ideal frequency response defined by the following equation:

$$I(\omega_1, \omega_2) = \begin{cases} 1 & , \omega_1^2 + \omega_2^2 \leq (0.4\pi)^2 \\ 0 & , \text{otherwise } -\pi \leq \omega_1, \omega_2 \leq \pi \end{cases}$$

   Construct an APL array with its values. Note that for compatibility with the FFT, the range on ω_1 and ω_2 should be zero to 2 rather than $-\pi$ to $+\pi$. Use a 32 by 32 point array.

2. Find the ideal impulse response $i(n1, n2)$ and plot it.

3. Choose one of the 1-D windows and plot it. Define a 2-D rectangular window from it. Multiply $i(n1, n2)$ by the window to get $h(n1, n2)$. The window should by 11 by 11 points in size.

4. Use the Fourier Transform (FFT) of $h(n1, n2)$ to get the filter frequency response. Plot this using both 3-D plotting and contour plotting.

SOLUTION

1. By using the function ILPFILT (Ideal LP filter) samples of the freauency response of an ideal lowpass filter is generated (32 by 32 point array). The plots achicved by using the surface plotting procedure in STATGRAPHICS can be seen in Figure D.44.

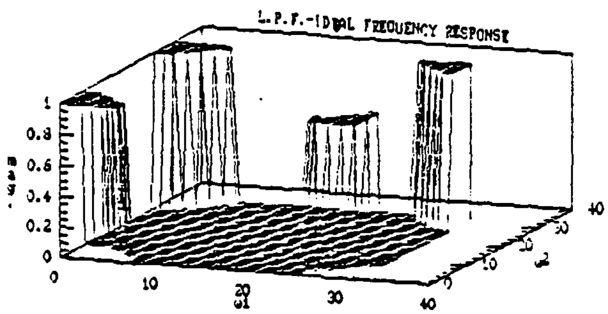
2. By using the IFFT2D function (Inverse 2-D Fast Fourier Transform) the ideal impulse response is founded and plotted in Figure D.45.

3. By using the function HAMMINGR we generate a 2-D HAMMING window with rectangular base. See Figure D.46.

4. Figure D.47 shows the product of the HAMMING window and the ideal impulse response.

5. By using the function FFT2D (2-D Fast Fourier Transform) the frequency response of the windowed filter is found, and by using the function XMAGN the magnitude of the result is calculated and plotted as in Figures D.48 and D.49 (contour plot). Note the use of the function SHIFT2D to show the frequency response in the interval 0 to 2π.

```
II←0.4×PI
N←32
WC←II ILPFILT N
WCU←SHIFT2D WC
```



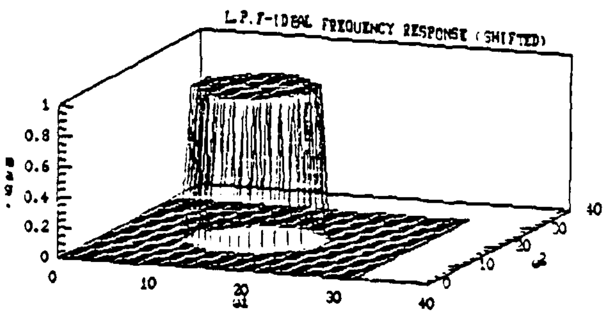
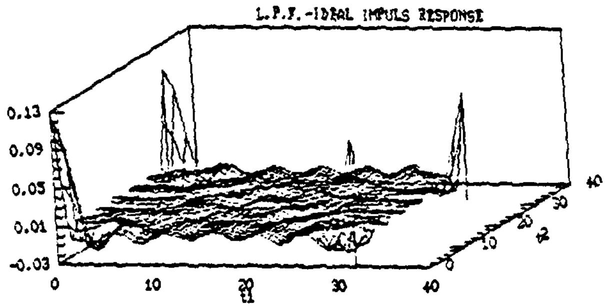


Figure D.44. Calculating and Plotting the Ideal Frequency Response.

```
WTU←IFFT2D WCU
WTUS←UNSHIFT2D WTU
```



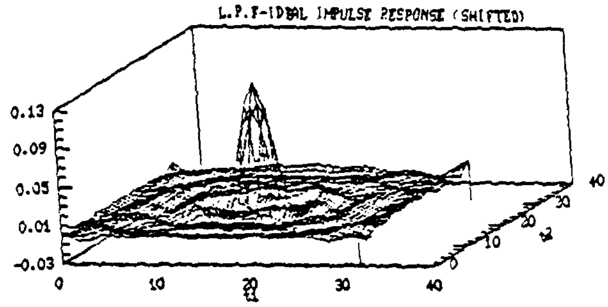


Figure D.45. Calculating and Plotting the Ideal Impulse Response.

```
II←11
N←32
WTR←II HAMMING2 N
```

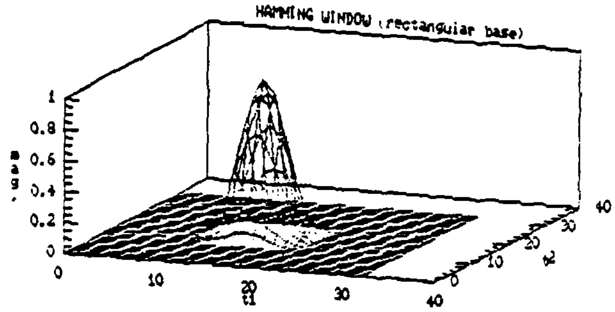


Figure D.46. Generating and Plotting the Hamming Window.

```
WTW←WTUS×WTR
```

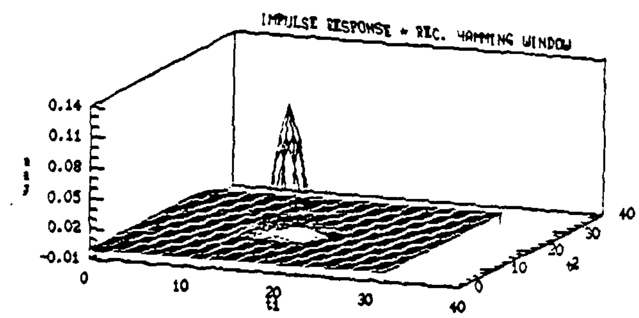


Figure D.47. The Product of the Hamming Window and the Ideal Impulse Response.

```
WTW←SHIFT2D WTW
WFW←FFT2D WTW
WF←XMAGN WFW
WFS←UNSHIFT2D WF
```

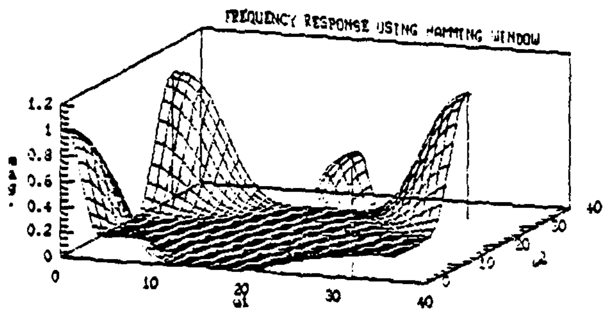


Figure D.48. The Design Frequency Response.

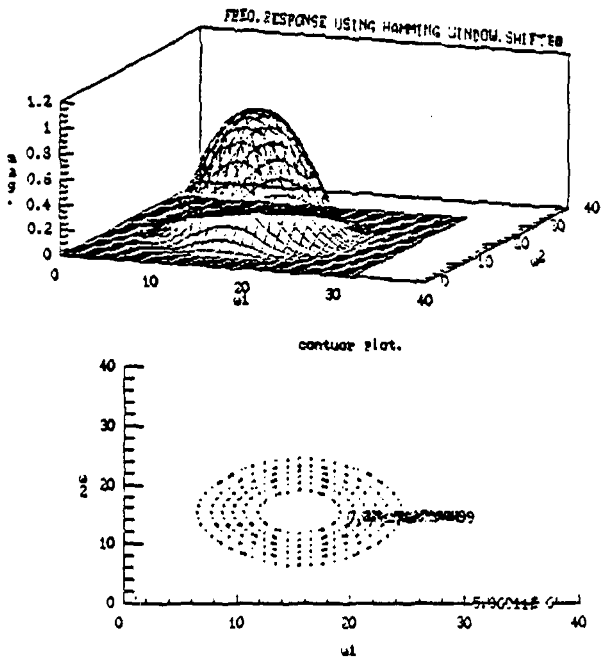


Figure D.49. The Shifted Design Frequency Response.

LIST OF REFERENCES

1. Gilman, Leonard and Rose, J. Allen, APL, An Interactive Approach, Third Edition, John Wiley and Sons, Inc., New York, 1983.

2. Iverson, Kenneth E., A Programming Language, John Wiley and Sons, Inc., New York, 1962.

3. STSC, STATGRAPHICS, Statistical Graphics System, Statistical Graphics Corporation, Users Guide, A PLUS*WARE Product, 1986.

4. Strum, Robert D. and Kirk, Donald E., First Principles of Discrete System and Digital Signal Processing, Addison-Wesley, Inc., Reading, MA, 1988.

5. Therrien, C. W., EC 3410 Class Notes, 1987.

6. Therrien, C. W., EC 4440 Class Notes, 1987.

7. STSC, APL*PLUS, System for PC, Reference Manual, A PLUS*WARE Product, STSC, 1986.

8. Orfanidis, J. Sophocles, Optimum Signal Processing: An Introduction, Macmillan Publishing Company, New York, 1984.

9. Dudgeon, Dan E. and Mersereau, Russell M., Multidimensional Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1984.

INITIAL DISTRIBUTION LIST

		No. Copies
1	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Professor John P. Powers, Code 62 Department Chairman Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
4.	Professor Charles W. Therrien, Code 62Ti Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	3
5.	Professor R. D. Strum, Code 62St Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
6.	Professor Donald E. Kirk Dean of Graduate Studies and Research School of Engineering San Jose State University One Washington Square San Jose, California 95192-0080	1
7.	Professor Murali Tummala Code 62Tu Naval Postgraduate School Monterey, California 93943-5000	1

8. Professor Paul Moose 1
Code 62Me
Naval Postgraduate School
Monterey, California 93943-5000

9. Yoel Katzir 3
36/A A.L. Yaffe
Hertzelia 46321, ISRAEL

10. Director, Research Administration, Code 012 1
Naval Postgraduate School
Monterey, California 93940-5000

11. Dr. Amnon Shefi 1
10 Mol St.
Netania 42540, ISRAEL

12. Moshe Marom 2
SMC # 1949
Naval Postgraduate School
Monterey, California 93943-5000

13. Roberto Ventura 1
Escalla Naval "Almirante Padilla"
Carfajena, Columbia

14. Yim, Jae Yong 1
SMC # 1216
Naval Postgraduate School
Monterey, California 93943-5000

15. Commander Naval Technical Training 1
Corry Station
Pensacola, Florida 32511-5000

16. Commander Naval Security Group 1
3801 Nebraska Avenue N.W.
Washington, DC 20390-0008

17. Space and Naval Warfare Command — 1
National Center # 1
2511 Jefferson Davis Highway
Arlington, Virginia 22202

END
DATE
FILMED
8-88
DTIC